1986

Intercalation properties of context-free languages

Rattikorn Boonyavatana
Iowa State University

8703689

Boonyavatana, Rattikorn

INTERCALATION PROPERTIES OF CONTEXT-FREE LANGUAGES

Iowa State University PH.D. 1986

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __√__.

1. Glossy photographs or pages ______
2. Colored illustrations, paper or print ______
3. Photographs with dark background ______
4. Illustrations are poor copy ______
5. Pages with black marks, not original copy ______
6. Print shows through as there is text on both sides of page ______
7. Indistinct, broken or small print on several pages __√__
8. Print exceeds margin requirements ______
9. Tightly bound copy with print lost in spine ______
10. Computer printout pages with indistinct print ______
11. Page(s) ______ lacking when material received, and not available from school or author.
12. Page(s) ______ seem to be missing in numbering only as text follows.
13. Two pages numbered ______. Text follows.
14. Curling and wrinkled pages ______
15. Dissertation contains pages with print at a slant, filmed as received ______
16. Other ______

Intercalation properties of context-free languages

by

Rattikorn Boonyavatana

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major : Computer Science

Approved :

Members of the Committee :

Signature was redacted for privacy.

~~In Charge of~~ Major Work

Signature was redacted for privacy.

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1986

TABLE OF CONTENTS

CHAPTER 1.

INTRODUCTION

Proving deeper properties of classes of languages is useful in at least two ways. First, our knowledge of the class in question has increased allowing deeper theoretical question be asked and answered. The second benefit becomes substantial when the particular property is easily testable ; it then becomes a convenient tool for proving that some languages are not in the class. Note that such properties give necessary conditions which usually are not sufficient. Among these properties are intercalation properties which sometimes are known as pumping lemmas or iteration theorems. Pumping lemmas provide a powerful tool for proving that certain languages do not belong to some given class of languages. The classic pumping lemmas were obtained by Rabin and Scott for regular languages [44], and by Bar-Hillel, Perles and Shamir for context-free languages [9]. The main advantage of such pumping lemmas is in the brevity and clarity of the associated proof methods. Because of their simplicity, many pumping lemmas for other families of languages were constructed in the same spirit, particularly for (nonregular) subfamilies of context-free languages. In 1968, Ogden [40] proved pumping lemmas for pushdown store and stack languages. Pumping conditions for deterministic languages, deterministic one-counter and deterministic ETOL languages were presented later by Harrison and Havel [28], Boasson [12] and Ehrenfeucht and Rozenberg [20] respectively. In recent years, more pumping lemmas were proved for strict deterministic context-free languages of a given degree, LL(k), simple precedence and real-time deterministic context-free languages [34, 10, 36, 32]. Kløve

[35] presented a study of more general pumping properties.

Because the family of context-free languages is central among the language families studied in formal language theory [27, 29, 11], with applications in compiler design techniques [1, 2, 45] and other areas [51, 14], it would be useful to know more about intercalation properties of context-free languages. In this research, we study the properties of pumping conditions for various subclasses of context-free languages and unify some of them in a more general form. Our first result, inspired by the classical pumping lemma of [9], is a pumping lemma for nonterminal bounded languages. This pumping lemma is a generalization of a pumping lemma for linear context-free languages (see ex. 6.11 [29]). In [15], we have strengthened these results by proving an Ogden-type pumping lemma in the spirit of [39].

Horvath [30] studied languages that satisfy the pumping lemma of [9], and showed that there are noncontext-free languages at all levels of the Chomsky hierarchy that satisfy the pumping lemma. Similar results regarding Ogden's lemma are proved in [13]. Likewise, we have produced languages which satisfy the pumping conditions (Ogden-type conditions) at various levels of the Chomsky hierarchy for the family of nonterminal bounded languages.

At this point, we began a deeper investigation of pumping conditions and became aware of other necessary conditions for context-free languages. It turned out that there are noncontext-free languages which satisfy (some or all of) these conditions, rendering (some or all of) these lemmas useless in those special cases.

Faced with this problem one can try to strengthen the existing pumping lemmas as Ogden [39] did to the pumping lemma of [9]. In 1976, Wise [50] obtained a strong characterization of context-free languages which has a flavor of pumping. (Similar conditions for regular languages were developed by Jaffe [33].) However, this approach seems to have a weakness as an applicative tool due to its rather involved application methodology. Thus, we will not be concerned with such strong conditions. In 1978, Sokolowski [49] proved another property of context-free languages which he showed to be applicable in some cases where the classical pumping lemma failed. Grant [24] extended the conditions of [49] to a considerably stronger condition. Finally, in 1982, Bader and Moura [7] strengthened Ogden's conditions by proving a generalized Ogden's lemma.

On the other hand, instead of strengthening an existing pumping lemma, one can try to prove a completely new necessary condition. In [46, 42], a new and interesting "interchange lemma" for context-free languages was proved. The novel feature of the interchange lemma is that new strings in the language are obtained by subword-interchanging between words in the language rather than by the "standard pumping". The first application of the interchange lemma was a nice solution to an open question regarding the language of repetitive strings [6]. Another application of the interchange lemma was supplied by Main [37] who showed that the language of permutation-containing strings (over any alphabet of at least 16 letters) is not context-free. In view of all these results, we study and compare the various pumping conditions with respect to their power. In [17], we have formulated the linear versions of these pumping conditions: the classic pumping condition [9], Ogden's condition [39], and the generalized Ogden's condition [7]. We have also formulated the linear interchange lemma [19] and then compared the interchange conditions to the above pumping conditions (both in context-free and linear cases) and to the Sokolowski-type conditions [49, 24].

The family of deterministic context-free languages is one of the most important classes of context-free languages. The main reason for that is in the area of parsing, compilation and translation [1, 2]. In 1985, Igarashi [32] presented three pumping lemmas for the family of real-time deterministic context-free languages. He showed several examples how his lemmas can be used. However, there was no known deterministic context-free language that is not real-time but that cannot be proved by these lemmas. We prove, by constructing appropriate counterexamples, that none of his conditions is sufficient.

In the following chapters we will discuss the above results in greater detail. Chapter 2 presents the basic definitions, terminology and includes the classical pumping conditions for context-free languages. Chapter 3 presents our pumping lemma and Ogden's lemma for nonterminal bounded languages. Also presented are a discussion on sufficiency and applications for both. In chapter 4 we compare the various pumping conditions (all of which are special cases of the generalized Ogden's condition of [7]) and the Sokolowski-type conditions for context-free languages [18]. We also present the linear analogues of these pumping conditions and conduct a comparison between the general pumping conditions and the linear pumping conditions [18]. Chapter 5 is devoted to the study of the interchange lemmas as presented in [19]. We prove the linear interchange lemma and then compare the interchange lemmas for both context-free and linear cases to various pumping conditions in both context-free and linear cases. Moreover, we compare the interchange conditions with the Sokolowski-type conditions [49, 24]. The relationships among classes of languages which satisfy these conditions are explored. In chapter 6, we present a short study of Igarashi's pumping conditions for real-time deterministic context-free languages. Chapter 7 contains a concluding summary.

CHAPTER 2.

BASIC DEFINITIONS AND SOME BACKGROUND

In this chapter we will bring together the basic, usually well-known, definitions, conventions and results. We mainly follow [1, 27, 29] for our terminology and notation.

Defining Grammars and Languages

Definition. A *context-free grammar (cfg)* is a construct $G = (N, T, P, S)$. N and T are two disjoint sets of *nonterminals* and *terminals* respectively ; P is a finite set of *productions* each of the form $A \rightarrow \alpha$ with A in N and α in $(N \cup T)^*$; the *start symbol* S is in N.

We will use V to denote $N \cup T$, the *vocabulary* of G. G_A , where A is a nonterminal of G, will be the grammar resulting from G by making A the start symbol; thus $G = G_S$. Let X be a subset of V and let w be in V^*. $\#_X(w)$ is the number of occurrences of symbols of X in w. Thus $\#_T(w)$ is the number of occurrences of terminals in w. $\#_V(w)$ is the *length* of w, also denoted by $| w |$. The language generated by G is denoted by L(G); a language L is a *context-free language (cfl)* if it is generated by some cfg.

A production $A \rightarrow \alpha$, in a cfg G, is *linear* if $\#_N(\alpha) \leqslant 1$. A cfg is *linear (lcfg)* if all its productions are linear. A cfl is *linear (lcfl)* if it is generated by some lcfg.

Let z be a (terminal) word of length n. An integer i, $1 \leqslant i \leqslant n$, is a *position* in z. For z we may regard some positions as *marked positions* or *distinguished posi-*

tions (dp's) and some as *excluded positions* (ep's). A particular position can be both a dp and an ep or, perhaps, neither. We will sometimes write $d\bar{e}p$ to mean a dp that is not an ep. The obvious convention that we use here can be extended to other combinations. d(z) and e(z) are respectively the number of dp's and ep's in z. A word with a marking defined on it is said to be a *marked word.* For a more precise definition of marking see [27].

Remark. When u and v are substrings of z then by d(uv) (and e(uv)) we mean the number of dp's (ep's) in z that occur within the substrings u and v.

Notation. (1) We will use capital letters for families of languages. Thus CFL and LIN are respectively the class of all cfl's and lcfl's.

(2) ϵ will denote the empty word; unit productions are productions of the form $A \rightarrow B$ where A, B are nonterminals; ϵ-production is a production of the form $A \rightarrow \epsilon$; height of a tree is the length of the longest root-to-leaf path in the tree.

(3) yield(t) denotes the string derived from the parse tree t.

Classic Pumping Conditions for CFL

The first, by now classic, pumping lemmas were obtained by Rabin and Scott for regular languages [44], and by Bar-Hillel, Perles and Shamir for context-free languages [9]. Intuitively, the classic pumping lemma of [9] states that from any word in the cfl language which is long enough, one can obtain other words in the language by deleting or by repeating some subwords of the given word an arbitrary number of times; thus, the adjectives "pumping" and "iteration". Ogden [39] strengthened the pumping lemma for context-free languages. The additional power of Ogden's lemma seems to stem from the use of marked positions which helps to cut down the numbers of factorizations considered in the pumping lemma. It is known that there exist noncontext-free languages satisfying pumping lemma [30] and Ogden's lemma [13] and hence, both of these conditions are not sufficient. Here we present the two pumping conditions for CFL mentioned above.

Pumping lemma for CFL [9] If L is a context free language then there is a constant n (depending only on L) such that if z is in L and $n \leqslant |z|$ then z can be written as z = uvwxy such that

1) $|vwx| \leqslant n$
2) $|vx| \geqslant 1$
3) for every $i \geqslant 0$, uv^iwx^iy is in L.

Ogden's lemma for CFL [40] If L is a context free language then there is a constant n (depending only on L) such that if z is in L and $d(z) \geqslant n$ then z can be written as z = uvwxy such that

1) $d(vwx) \leqslant n$,
2) $d(vx) \geqslant 1$,
3) for every $i \geqslant 0$, uv^iwx^iy is in L.

CHAPTER 3.

PUMPING CONDITIONS FOR NTBL

In this chapter we prove a pumping lemma and an Ogden-type pumping lemma for nonterminal bounded languages. These are languages generated by nonterminal bounded grammars, see [3, 4, 8, 22, 27]. Alternately, these are languages accepted by finite-turn pushdown automata, see [23]. Nonterminal bounded languages are sometimes called ultralinear, [47] but compare [23]. Our pumping lemma is a generalization of a pumping lemma for linear context-free languages (see exercise 6.11 of [29]). Similarly, an Ogden-type pumping lemma is a generalization of an Ogden's lemma for linear context-free languages (see proposition 6.6, section v.6 of [11]).

This chapter contains three sections. The first section gives preliminary definitions and the proof of the pumping lemma and Ogden's lemma for linear languages. The second section contains our main results, the pumping lemma and the Ogden's lemma for nonterminal bounded languages together with examples of their application. In the third section we give counterexamples to show that none of these lemmas provides a sufficient condition. In fact, we construct counterexamples at various levels of the Chomsky hierarchy, each of which satisfies the conditions of our pumping conditions.

Preliminaries

In this section more basic definitions and notation related to the material in this chapter are introduced. We will discuss pumping lemma and Ogden's lemma for linear context free languages at the end of this section with an eye towards a generalization to be discussed in subsequent sections.

Let t be a derivation tree for some marked word in language L = L(G) for some cfg G. We define a node n of t to be a *branch node* if n has at least 2 direct descendants both of which have marked descendants. Let q be a leaf of a particular root-to-leaf path π of t. Then a branch node n on π is a *left branch node* (relative to π) if a direct descendant of n not on the path π has a marked descendant to the left of q ; otherwise, n is a *right branch node.* It should be noted that, contrary to the usual definition (cf. Ogden's original proof [39]) the notions of left and right branch nodes are not symmetrical i.e., here, a branch node cannot be simultaneously left and right.

We now define the rank functions. Let G = (N, T, P, S) be a cfg and let α in V^* be a sentential form. If the set $\{\ \#_N(\beta) \mid \alpha \overset{*}{\underset{G}{\Rightarrow}} \beta\ \}$ is finite then we let rank(α) = max $\{\ \#_N(\beta) \mid \alpha \overset{*}{\underset{G}{\Rightarrow}} \beta\ \}$ otherwise rank(α) is undefined. A cfg G for which rank(A) is defined for every nonterminal A is called *nonterminal bounded (ntbg).* The *rank of G*, rank(G), is max[rank(A)] where A is in N. G is *k-nonterminal bounded (k-ntbg)* if rank(G) = k. L is k-nonterminal bounded (k-ntbl) if it is generated by some k-ntbg. Note that rank(w) = 0 for w in T^* and for $\alpha = \alpha_1\alpha_2\ldots\alpha_k$ in V^*, rank(α) = $\sum_{i=1}^{k}$rank(α_i) ; moreover, observe that every ntbg has nonterminals of rank 1 and that "1-ntbl" and "linear" are synonymous terms.

Example. The cfl L = $\{\ a^nb^n \mid n \geqslant 1\ \}$ is generated by the cfg G : S $\rightarrow$ ab, S $\rightarrow$ aSb.

Clearly rank(G) = rank(S) = 1.

Let $k \geqslant 1$ be an integer and define $L_k = L^k$ (i.e., $L \cdot L \ldots L$, k times). Then L_k is generated by the cfg $S \rightarrow AA \ldots A$ (k times), $A \rightarrow ab$, $A \rightarrow aAb$; now, $\mathrm{rank}(A) = 1$, $\mathrm{rank}(G) = \mathrm{rank}(S) = k$.

We will use p to denote the maximum number of occurrences of terminals in the productions of a grammar, i.e., $p = \max \{ \#_T(\alpha) \mid A \rightarrow \alpha$ is a production in G $\}$.

In the rest of this section we will prove a pumping lemma and an Ogden's lemma for linear languages. They will serve as a basis for the generalization in the following sections.

Lemma 3.1. (Ogden's lemma for LIN) If L is a lcfl then there is a constant n (depending only on L) such that if z is in L and $d(z) \geqslant n$ then z can be written as $z = uvwxy$ such that

1) $d(uvxy) \leqslant n$,
2) either each of u, v, w or each of w, x, y contains a marked position,
3) for every $i \geqslant 0$, uv^iwx^iy is in L.

It is perhaps appropriate to point out here that Ogden's lemmas for CFL and LIN differ precisely in condition (1) of the above lemma where in the CFL-case the requirement is $d(vwx) \leqslant n$ rather than $d(uvxy) \leqslant n$. We should also mention that in the classical pumping lemma for linear languages condition (1) reduces to $|uvxy| \leqslant n$ [11, section v.6, proposition 6.6].

We will first prove a claim which relates the number of marked positions of a word with its derivation tree in a grammar.

Claim 3.1. Let G be a lcfg without ϵ or unit productions. Let t be the derivation tree for the derivation $A \overset{*}{\underset{G}{\Rightarrow}} z$, z in T^*, and C a root-to-leaf path in t with maximum number of branch nodes. If C has $\leqslant$b branch nodes, then d(yield(t)) $\leqslant$ bp+1.

Proof of claim 3.1. By induction on b. For b = 0, it is clear by the definition of branch node that d(yield(t)) $\leqslant$ 1. Now let t be as described in the claim (with b $\geqslant$ 1). Let B be the label of the first branch node on the path C. Since G is a lcfg, the first step in the derivation from B is $B \Rightarrow \alpha_1 B_1 \beta_1$ where α_1, β_1 are in T^* and hence t is as in figure 1. The part of the path C which is in t_1 has $\leqslant$b−1

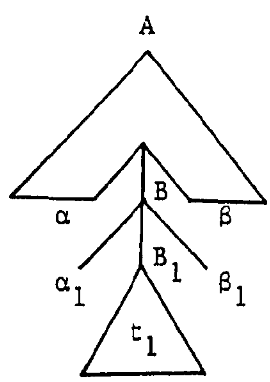

Figure 1 : Derivation tree t

branch nodes and $d(\alpha) = d(\beta) = 0$. By induction hypothesis, $d(yield(t_1)) \leqslant (b-1)p + 1$ and so $d(yield(t)) = d(\alpha_1) + d(\beta_1) + d(yield(t_1)) \leqslant p + (b-1)p + 1 =$

bp + 1. We have proved the claim. □

Proof of lemma 3.1. Without loss of generality we may assume that ϵ is not in L and L = L(G) where G = (N, T, P, S) has no ϵ or unit productions. Put k = | N | (cardinality of N), and n = 2(k+1)p + 2. Let z be in L with $d(z) \geqslant n$ and let t be the derivation tree for z in G; let C be the root-to-leaf path in t with maximum number of branch nodes.

By claim 3.1, C has at least 2k+3 branch nodes. Let $b_1, b_2, \ldots, b_{2k+3}$ be the first 2k+3 branch nodes in the path C. We may assume that at least k+2 of $b_1, \ldots, b_{2k+3}$ are left branch nodes. The other case can be treated analogously. Let $l_1, \ldots, l_{k+2}$ be the first k+2 left branch nodes in the sequence $b_1, \ldots, b_{2k+3}$. Since there are k non-terminals we can find two nodes among $l_2, \ldots, l_{k+2}$, say l_e and l_f such that (1) l_e and l_f are labeled by the same nonterminal, say A, and (2) l_e is an ancestor of l_f. This situation is shown in figure 2.

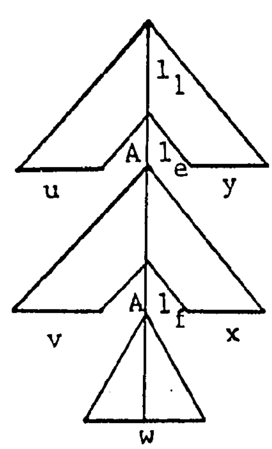

Figure 2 : Derivation tree for z

Since l_f is an ancestor of b_{2k+3}, the path along C from the root of t down to but excluding l_f has at most 2(k+1) branch nodes. By claim 3.1, $d(uvxy) \leqslant 2(k+1)p + 1 < n$. Since l_1, l_e and l_f are left branch nodes, each of u, v and w has at least one marked position. Hence, condition (2) of lemma 3.1 is satisfied. Finally, we have $S \overset{*}{\underset{G}{\Rightarrow}} uAy$, $A \overset{*}{\underset{G}{\Rightarrow}} vAx$ and $A \overset{*}{\underset{G}{\Rightarrow}} w$. Therefore, $S \overset{*}{\underset{G}{\Rightarrow}} uv^iwx^iy$ for all $i \geqslant 0$. The proof of the lemma is now complete. □

By defining $d(z) = |z|$ for all z in L, we can easily obtain a pumping lemma for LIN.

Corollary 3.1. (Pumping Lemma for LIN) If L is a lcfl then there is a constant n (depending only on L) such that if z is in L and $n \leqslant |z|$ then z can be written as $z = uvwxy$ such that

1) $|uvxy| \leqslant n$
2) $|vx| \geqslant 1$
3) for every $i \geqslant 0$, uv^iwx^iy is in L.

Example 1. The language $L = \{ a^ib^{i+j}a^j \mid i, j \geqslant 1 \}$ is not linear. To see this let n be the constant of the pumping lemma for LIN and consider $z = a^nb^{2n}a^n$. Then z cannot be written as uvwxy with the conditions of corollary 3.1 fulfilled.

Example 2. Let $\Sigma = \{a_1, b_1, c_1\}$. We will use lemma 3.1 to show that $L = b\Sigma^* \cup \{ a^nba_1^kb_1^{k+m}c_1^mc^n \mid k, m, n \geqslant 1 \}$ is not lcfl but L satisfies a pumping lemma for lcfl. We will first show that L satisfies the linear pumping conditions with constant 7. Let z be in L, $|z| \geqslant 7$. Then either $z = a^iba_1^jb_1^{j+k}c_1^kc^i$

for some i, j, k $\geqslant 1$ or $z = bz_1$ with z_1 in Σ^+. In the second case we may factor $z = uvwxy$ where $u = b$, v = the first symbol of z_1, w = the rest of z and $x = y = \epsilon$. In the first case let $u = y = \epsilon$, v = the first a of z, x = that last c of z, w = the rest of z. In both cases, it is easy to see that $|uvxy| \leqslant 7$, $|vx| \geqslant 1$ and uv^iwx^iy is in L for all $i \geqslant 0$. We will now show that L is not lcfl. Suppose L is lcfl and let n be the Ogden's lemma constant for L. Consider $z = a^nba_1^nb_1^{2n}c_1^nc^n$ where all positions of $a_1^nb_1^{2n}c_1^n$ are marked. Clearly, $d(z) > n$ and thus, we can factor $z = uvwxy$ such that the three conditions of lemma 3.1 hold. It is obvious that v and x can contain only one type of letter and, to satisfy condition (2), at least one of these is within $a_1^nb_1^{2n}c_1^n$. Furthermore, because of condition (1), neither v nor x can contain occurrences of b_1 which implies immediately that pumping of z will yield strings out of L. Thus condition (3) of lemma 3.1 fails, showing that L is not lcfl.

Pumping Lemma and Ogden's Lemma for NTBL

We will now prove an Ogden's lemma for nonterminal bounded languages which is a generalization of lemma 3.1. As for LIN, a pumping lemma for NTBL is easily obtained from the Ogden's lemma. We will first prove an auxiliary claim analogous to claim 3.1, then present the proof of the main lemma.

Claim 3.2. Let G be a ntbg and t be the derivation tree for a derivation $A \overset{*}{\underset{G}{\Rightarrow}} z$, z in T^*, where rank(A) = r. If C, a root-to-leaf path in t with maximum number of branch nodes, has at most b branch nodes, then $d(z) \leqslant (rb-r+1)p + r$.

Proof. By induction on r. For $r = 1$ we have the linear case which has been treated in claim 3.1. Now let $r \geqslant 2$ and suppose the claim holds for nonterminals of rank $< r$. Let A, G, C, z, t and b be as stated. Starting from the root of t, let n_1(labeled by A_1) be the first branch node on the path C. The first step in the derivation from A_1 is $A_1 \Rightarrow \alpha_1 B_1 \alpha_2 B_2 \ldots \alpha_n B_n \alpha_{n+1}$ ($n \geqslant 1$) and we have the situation as shown in figure 3, where for all i, α_i is in T^*, $\text{rank}(B_i) = r_i$, $\sum_{j=1}^{n} r_j \leqslant$

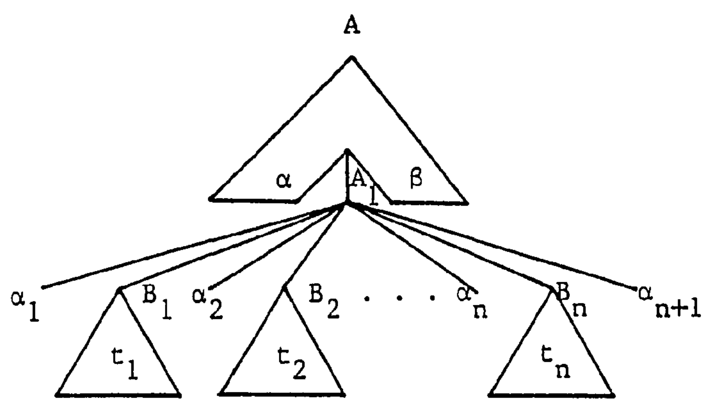

Figure 3 : Derivation tree t

$\text{rank}(A_1) \leqslant \text{rank}(A) = r$, t_i has at most $b-1$ branch nodes and $z = \alpha\alpha_1 \text{yield}(t_1) \ldots \text{yield}(t_n)\alpha_{n+1}\beta$. Note that α and β have no marked positions. We want to show that $d(z) \leqslant (rb-r+1)p + r$. There are two cases to be considered.

Case 1 : $n > 1$. We have $r_i < r$ ($i = 1, \ldots, n$). By induction hypothesis $d(\text{yield}(t_i)) \leqslant (r_i(b-1) - r_i + 1)p + r_i$. Thus

$$
\begin{aligned}
d(z) &= \sum_{i=1}^{n+1} d(\alpha_i) + \sum_{i=1}^{n} d(\text{yield}(t_i)) \\
&\leqslant p + \sum_{i=1}^{n} (r_i(b-1) - r_i + 1)p + \sum_{i=1}^{n} r_i
\end{aligned}
$$

$$\leqslant \quad p + \sum_{i=1}^{n} r_i(b-1)p + r$$

$$\leqslant \quad p + r(b-1)p + r = (\ rb-r+1\ )p + r$$

Case 2 : $n = 1$. We argue inductively on b. If $b = 1$, A_1 is the only branch node on C, $d(yield(t_1)) \leqslant 1$ and so $d(z) \leqslant p + 1 < (rb-r+1)p + r$. Now assume that $b \geqslant 2$ and the claim holds for C with less than b branch nodes. By induction hypothesis, $d(yield(t_1)) \leqslant (\ r_1(b-1) - r_1 + 1)p + r_1$. Thus, $d(z) \leqslant p + d(yield(t_1)) \leqslant (r_1 b - r_1 + 1)p + r_1 \leqslant (rb - r + 1)p + r$. This completes the proof of the claim. □

Theorem 3.1 (Ogden's lemma for NTBL). If L is an r-ntbl and generated by a r-ntbg $G = (\ N, T, P, S\ )$ then there exists a constant n (depending on L) such that if z is in L and $d(z) \geqslant n$ then z can be written as $z = z_1 z_2 \ldots z_s$, for $1 \leqslant s \leqslant r$, where each z_i can be written as $z_i = u_i v_i w_i x_i y_i$ such that

1) $\sum_{i=1}^{s} d(u_i v_i x_i y_i) \leqslant n$,

2) either each of u_i, v_i, w_i or each of w_i, x_i, y_i contains a marked position,

3) for all natural numbers $a_i \geqslant 0$ ($1 \leqslant i \leqslant s$) $z_1^{(a_1)} z_2^{(a_2)} \ldots z_s^{(a_s)}$ is in L where $z_i^{(b)} = u_i v_i^b w_i x_i^b y_i$.

Proof. By induction on r. For $r = 1$, L is lcfl and we have proved the result in lemma 3.1. Let L be r-ntbl and $G = (\ N, T, P, S\ )$ an r-ntbg such that $L = L(G)$, $r \geqslant 2$, $k = |N| = \sum_{i=1}^{r} k_i$ where k_i is the number of nonterminals of rank i in G. Put $n = (2rk+r+1)p + r + 1$. Consider a derivation tree t for z in L(G) such that

$d(z) \geqslant n$. Let C be root-to-leaf path in t with maximum number of branch nodes. By claim 3.2, the path C has at least 2k+3 branch nodes on it. Let $b_1, b_2, \ldots, b_{2k+3}$ be the first (topmost) 2k+3 branch nodes on C. Among these there are at least k+2 left branch nodes or at least k+2 right branch nodes. Since there are k nonterminals, there exist branch nodes b_f, b_i and b_j, $1 \leqslant f < i < j$, each of which is to be as high as possible in C and such that

(1) b_f, b_i and b_j are of the same type , i.e., either all are left or all are right branch nodes (note that all branch nodes above b_f must be of type different from b_f),

(2) b_i and b_j are labeled by the same nonterminal, say A,

(3) there is at most one nonterminal $B \neq A$, for which there are four (or three) proper ancestor branch nodes of b_j labeled by B (two of each type), and

(4) for each nonterminal E distinct from A and the nonterminal B of (3), there can be at most two ancestor branch nodes of b_j labeled by E (one of each type).

It is left to the reader to convince himself that such branch nodes can in fact be found. We will consider two cases depending on the form of the derivation tree from the root to b_j.

Case 1 : The derivation from S (the root) to A (b_j) uses only linear productions. Since b_j is an ancestor of b_{2k+3}, the path along C from S down to A (b_j) excluding b_j contains at most 2(k+1) branch nodes. By the proof of lemma 3.1, we can write

$z = uvwxy$ such that (1) $d(uvxy) \leqslant 2(k+1)p + 1 \leqslant n$, (2) either each of u, v and w or each of w, x and y contains a marked position, and (3) for every $i \geqslant 0$, uv^iwx^iy is in L. Thus, the theorem is obtained with $s = 1$.

Case 2 : The nodes b_i, b_j occur after a nonlinear production : $B \to \alpha_1 B_1 \alpha_2 B_2 \ldots \alpha_m B_m \alpha_{m+1}$, $2 \leqslant m \leqslant r'$, where $\text{rank}(B) = r' \leqslant r$, $\text{rank}(B_i) = r_i$, α_i in T^* and $\sum_{i=1}^{m} r_i \leqslant r'$. This is illustrated in figure 4. The case where B is at the root of the tree (i.e., S = B) is easy and left to the reader; in what follows we assume $S \neq B$.

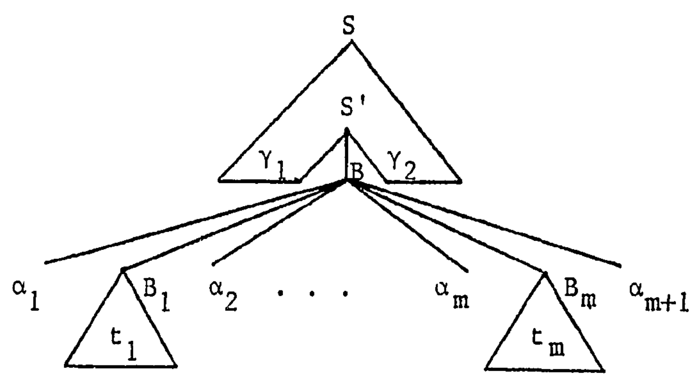

Figure 4 : Derivation tree for z where the derivation from B is the topmost nonlinear derivation step

From the way b_i and b_j are chosen it follows that the number of branch nodes on the path from S to B is $\leqslant 2\sum_{i=r'}^{r} k_i + 2$. By linearity of the derivation from S to S' each branch node on the path from S to S' can contribute at most p marked positions. Thus, we have $d(\gamma_1\gamma_2) + \sum_{i=1}^{m+1} d(\alpha_i) \leqslant 2\sum_{i=r'}^{r} k_i p + 2p$. Since $r \geqslant 2$ and

$\sum_{i=r'}^{r} k_i \geqslant 2$ it is easy to check that $2\sum_{i=r'}^{r} k_i p + 2p \leqslant 2\sum_{i=r'}^{r} k_i rp - 2rp + 2p$. We now consider d(yield(t_i)) for each subtree t_i. Put $n_i = (2r_i \sum_{j=1}^{r_i} k_j + r_i + 1)p + r_i + 1$ for $1 \leqslant i \leqslant m$.

Claim. There exists $1 \leqslant i \leqslant m$ such that $d(\text{yield}(t_i)) \geqslant n_i$.

Proof of Claim. Suppose that $d(\text{yield}(t_i)) < n_i$ for all i. Then

$$\begin{aligned}
d(z) &= d(\gamma_1\gamma_2) + \sum_{i=1}^{m+1} d(\alpha_i) + \sum_{i=1}^{m} d(\text{yield}(t_i)) \\
&< d(\gamma_1\gamma_2) + \sum_{i=1}^{m+1} d(\alpha_i) + \sum_{i=1}^{m} n_i \\
&\leqslant 2\sum_{i=r'}^{r} k_i rp - 2rp + 2p + \sum_{i=1}^{m}[(2r_i \sum_{j=1}^{r_i} k_j + r_i + 1)p + r_i + 1] \\
&\leqslant 2\sum_{i=r'}^{r} k_i rp - rp + 2p + 2p\sum_{i=1}^{m} r_i \sum_{j=1}^{r'-1} k_j + mp + r + m \\
&\leqslant 2\sum_{i=r'}^{r} k_i rp + 2r + 2p + 2\sum_{i=1}^{r'-1} k_i rp \leqslant 2krp + p + r + rp + 1 = n
\end{aligned}$$

This contradicts the assumption that $d(z) \geqslant n$. Hence, the claim is proved.

Let $J = \{ j \mid d(\text{yield}(t_j)) \geqslant n_j \}$; by the above claim, $1 \leqslant |J| \leqslant r$. By induction hypothesis, for each j in J, we can write $\text{yield}(t_j) = z_{j1}z_{j2}\ldots z_{js_j}$ where $1 \leqslant s_j \leqslant r_j$ and each z_{ji} can be written as $z_{ji} = u_{ji}v_{ji}w_{ji}x_{ji}y_{ji}$ with

1) $\sum_{i=1}^{s_j} d(u_{ji}v_{ji}x_{ji}y_{ji}) \leqslant n_j$

2) either each of u_{ji}, v_{ji} and w_{ji} or each of w_{ji}, x_{ji} and y_{ji} contains a marked position for all $1 \leqslant i \leqslant s_j$.

3) $z_{j1}^{(a_1)} z_{j2}^{(a_2)} \ldots z_{j s_j}^{(a_{s_j})}$ is in $L(G_{B_j})$.

Now we will write z (in figure 4) in the form $z = z_1 z_2 \ldots z_s$, $1 \leqslant s \leqslant r$ where z_i's satisfy the conditions stated in the theorem. We will argue in three cases :

Case 1 : The first z_1 is defined to be $u_1 v_1 w_1 x_1 y_1$ where $u_1 = \gamma_1 \alpha_1 \text{yield}(t_1) \ldots \alpha_h u_{h1}$;

$v_1 = v_{h1}$; $w_1 = w_{h1}$; $x_1 = x_{h1}$; $y_1 = y_{h1}$ where $h = \min J$.

Case 2 : The last z_s is defined to be $u_s v_s w_s x_s y_s$ where $u_s = u_{l s_l}$; $v_s = v_{l s_l}$;

$w_s = w_{l s_l}$; $x_s = x_{l s_l}$ $y_s = y_{l s_l} \alpha_{l+1} \text{yield}(t_{l+1}) \ldots \alpha_{m+1} \gamma_2$ where $l = \max J$.

Case 3 : For the interior z_g , $1 < g < s$, there are two possible cases. The first is when $z_g = z_{ji}$ for j in J and $1 < i < s_j$. For the second case, let h and q be consecutive elements of J, i.e., $h < q$ and for $h < i < q$, i is not in J. We then define $z_g = u_g v_g w_g x_g y_g$ where $u_g = u_{hs_h}$; $v_g = v_{hs_h}$; $w_g = w_{hs_h}$; $x_g = x_{hs_h}$;

$y_g = y_{hs_h} \alpha_{h+1} \text{yield}(t_{h+1}) \ldots \alpha_q$.

We have obtained $z = z_1 z_2 \ldots z_s$ ($s \geqslant 1$). Moreover, $s = \sum_{j \in J} s_j \leqslant \sum_{j \in J} r_j \leqslant \sum_{j=1}^{m} r_j \leqslant r$. Thus $1 \leqslant s \leqslant r$. It remains to show that all the conditions of the theorem hold. By induction hypothesis and the definition of J we have :

$$\begin{aligned}
\sum_{i=1}^{s} d(u_i v_i x_i y_i) &= d(\gamma_1 \gamma_2) + \sum_{i=1}^{m+1} d(\alpha_i) + \sum_{j \notin J} d(\text{yield}(t_j)) + \sum_{j \in J} \sum_{i=1}^{s_j} d(u_{ji} v_{ji} x_{ji} y_{ji}) \\
&< d(\gamma_1 \gamma_2) + \sum_{i=1}^{m+1} d(\alpha_i) + \sum_{j \notin J} n_j + \sum_{j \in J} n_j \\
&= d(\gamma_1 \gamma_2) + \sum_{i=1}^{m+1} d(\alpha_i) + \sum_{j=1}^{m} n_j
\end{aligned}$$

But this sum is $< n$ by the proof of the claim shown before. Thus, condition (1) is satisfied. By construction of the z_i's conditions (2) and (3) are obtained immediately. Hence the proof of theorem 3.1 is complete. □

The following pumping conditions are obtained from the Ogden's conditions by replacing any mention of the mapping d (and marked positions) by the mapping $|\cdot|$ (and the concept of length).

Corollary 3.2. (Pumping lemma for NTBL). If L is an r-ntbl then there is a constant n (depending only on L) such that if z is in L and $n \leqslant |z|$ then z can be written as $z = z_1z_2 \ldots z_s$, $1 \leqslant s \leqslant r$ where each z_i can be written as $z_i = u_iv_iw_ix_iy_i$ such that

1) $\sum_{i=1}^{s} |u_iv_iw_ix_iy_i| \leqslant n$

2) $|v_ix_i| \geqslant 1 \quad i = 1, \ldots ,s$

3) for all natural numbers $a_i \geqslant 0$ ($1 \leqslant i \leqslant s$)

$$z_1^{(a_1)}z_2^{(a_2)} \ldots z_s^{(a_s)} \text{ in L where } z_i^{(b)} = u_iv_i^bw_ix_i^by_i.$$

We will now give two examples in which theorem 3.1 is applied. In particular we show that the Ogden's condition of theorem 3.1 is stronger than the pumping condition of corollary 3.2.

Example 1. We will use the pumping lemma for NTBL to show that the language $L = \{ a_1^{n_1}b_1^{n_1}a_2^{n_2}b_2^{n_2} \ldots a_{k+1}^{n_{k+1}}b_{k+1}^{n_{k+1}} \mid \text{for } 1 \leqslant i \leqslant k+1,\ n_i \geqslant 1\}$ is not k-ntbl. Suppose L is k-ntbl and let n be the corresponding constant of the lemma. Consider the word $z = a_1^nb_1^na_2^nb_2^n \ldots a_{k+1}^nb_{k+1}^n$ in L. Clearly, $|z| > n$

and so by the pumping lemma for NTBL z may be written as $z = z_1 z_2 \ldots z_s$ with $1 \leqslant s \leqslant k$, where $z_i = u_i v_i w_i x_i y_i$ such that the three conditions of theorem 3.1 hold. Obviously, for each i, neither v_i nor x_i can contain two types of letters and moreover, since $v_i x_i \neq \epsilon$, neither of v_i, x_i can be ϵ. Now, z consists of 2k+2 different letters while at the same time it contains at most 2k pieces that get pumped, namely the v_i's and x_i's. It follows that at least two letters do not participate in the "pumping festivities" and that these letters come in pairs a_i, b_i. In other words, it cannot be that a_i does not get pumped while b_i does. Consider where such an unpumped piece $a_i^n b_i^n$ can be located within z; clearly, the only possible places are (i) within $y_j u_{j+1}$ for some $1 \leqslant j < s$ or (ii) within w_j for some $1 \leqslant j \leqslant s$. The first case is impossible because then $| y_j u_{j+1} | \geqslant 2n$ contradicting condition (1) of the pumping lemma. In the second case, $a_i^n b_i^n$ is within z_j and then v_j must consist of a_e's or b_e's for some $e < i$ while x_j must consist of a_f's or b_f's for some $i < f$. Since v_j and x_j are pumped together this situation is clearly impossible. We conclude that L is not k-ntbl. Note that L is (k+1)-ntbl.

Example 2. Let $\Sigma = \{ a_1, b_1, a_2, b_2, \ldots, a_{r+1}, b_{r+1} \}$. We will use theorem 3.1 to show that $L = \{ a^n b a_1^{n_1} b_1^{n_1} a_2^{n_2} b_2^{n_2} \ldots a_{r+1}^{n_{r+1}} b_{r+1}^{n_{r+1}} c^n \mid n, n_i \geqslant 1, i = 1, \ldots, r+1 \}$ $\cup\ b\Sigma^*$, is not r-ntbl but L satisfies the pumping lemma for r-ntbl. Similarly to example 2 in previous section, we can prove that L satisfies the pumping lemma for LIN. Clearly, any language that satisfies the linear pumping conditions also satisfies the pumping conditions for NTBL at any rank. Now we will use theorem 3.1 to show that L is not r-ntbl. Suppose L is r-ntbl and let n be the constant corresponding to L in theorem 3.1. Consider $z = a^n b a_1^n b_1^n a_2^n b_2^n \ldots a_{r+1}^n b_{r+1}^n c^n$ where all positions of a subword $z' = a_1^n b_1^n a_2^n b_2^n \ldots a_{r+1}^n b_{r+1}^n$ are marked, and thus $d(z) > n$. By theorem 3.1, z may be written as $z_1 z_2 \ldots z_s$ with $1 \leqslant s \leqslant r$, where $z_i = u_i v_i w_i x_i y_i$ such that the three conditions of that theorem hold. For each i, each of v_i and x_i can contain only one type of letter and at least one of them is contained in z' to satisfy condition (2). Now z' consists of 2r+2 distinct letters whereas only at most 2r pieces of v_i's and x_i's get pumped. This implies that there exist at least 2 letters that do not get pumped and they must come in pairs of a_i and b_i to keep the correct balance. These pairs of $a_i^n b_i^n$ can be placed either (1) within $y_j u_{j+1}$ for some $1 \leqslant j < s$ or (2) within w_j for some $1 \leqslant j \leqslant s$. In the first case we have $d(\ y_j u_{j+1}\ ) \geqslant 2n$ which violates condition (1) of the theorem. In the second case, $a_i^n b_i^n$ is within z_j and then v_j must consist of a_e's or b_e's for some $e < i$ while x_j must consist of a_f's or b_f's for some $i < f$. Clearly, this violates condition (3) of the theorem because v_j and x_j are pumped together. Thus L is not r-ntbl. Again note that L is (r+1)-ntbl.

Let NTBL(k), OL(k) and PL(k) denote classes of k-ntbl's, languages satisfying Ogden's condition (of theorem 3.1) and pumping condition (of corollary 3.2) with rank k, respectively. NTBL, OL_{ntbl} and PL_{ntbl} denote classes of ntbl's, the union of OL(k) and PL(k) for $k \geqslant 1$, respectively. We can obtain the inclusion relationships among these classes of languages as shown in the diagram of figure 5. (Note NTBL(1) = LIN, OL(1) and PL(1) will be referred to as OL_l and PL_l in the following chapters.)

The inclusion relationships of the diagram below follow from the definitions ; i.e., NTBL(1) $\subseteq$ NTBL(2) $\subseteq \ldots \subseteq$ NTBL, OL(1) $\subseteq$ OL(2) $\subseteq \ldots \subseteq$ OL_{ntbl} , PL(1) $\subseteq$ PL(2) $\subseteq \ldots \subseteq$ PL_{ntbl} , NTBL $\subseteq$ OL_{ntbl} $\subseteq$ PL_{ntbl} and for each k $\geqslant$ 1,

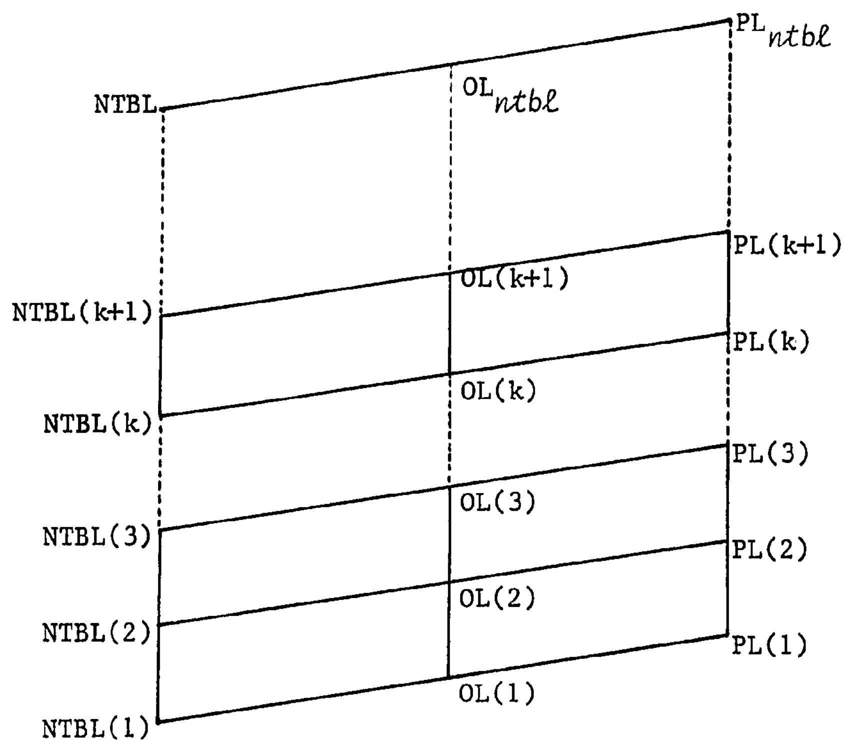

Figure 5 : The relationship among NTBL(k), OL(k) and PL(k)

NTBL(k) $\subseteq$ OL(k) $\subseteq$ PL(k). It remains to establish the strictness of the above inclusion relationships. Let k $\geqslant$ 1 be any integer. In example 1, we constructed a language L which is not in PL(k) and thus not in NTBL(k) and not in OL(k). On the other hand L is (k+1)-ntbl and thus in PL(k+1) and OL(k+1). Hence NTBL(k) $\subsetneq$ NTBL(k+1), OL(k) $\subsetneq$ OL(k+1) and PL(k) $\subsetneq$ PL(k+1). In example 2, we presented a language L which is not in OL(k) but in PL(1). Thus L is in

PL(k) and hence OL(k) $\subsetneqq$ PL(k). In [18] (see also the fourth section of chapter 4), a language $L = \{ z \text{ in } \{a,b\}^* \mid z = ab^p \Rightarrow p \text{ prime} \}$ was shown to be in OL(1) but not context-free. This shows that NTBL(k) $\subsetneqq$ OL(k). Hence, we establish the properness of the inclusion hierarchy of figure 5.

Languages Satisfying Pumping and Ogden-type Conditions

We have seen that lemma 3.1 and corollary 3.1 provide necessary conditions for linear languages whereas theorem 3.1 and corollary 3.2 provide necessary conditions for nonterminal bounded languages. We will now consider the question whether these conditions are also sufficient. More precisely, we will present counterexamples which show that these conditions are not sufficient. We start by counterexamples, inspired by [30], for pumping conditions followed by those for Ogden-type pumping conditions.

<u>For pumping lemma</u>

Let Σ be an alphabet consisting of at least two symbols and let a, b and c be symbols not in Σ. Let H be a language over Σ (with ϵ in H). Define

$$L_H = \{ a^n b w c^n \mid n \geqslant 1, w \text{ in } H \} \cup b\Sigma^*.$$

It is easy to see that choosing H to be properly cfl (i.e., cfl but not lcfl) makes L_H properly cfl. Likewise, choosing H to be properly context-sensitive (respectively : recursive, recursively enumerable, not recursively enumerable) makes L_H properly context-sensitive (respectively : recursive, recursively enumerable, not recursively enumerable).

We will now prove that L_H satisfies the pumping condition of corollary 3.1 with $n = 3$. Let z be in L_H, $|z| \geqslant 3$. Then either $z = a^i b z_1 c^i$ for some $i \geqslant 1$ and z_1 in H, or $z = bz_1$ with z_1 in Σ^+. In the latter case we may factor $z = uvwxy$ where $u =$ b, $v =$ the first symbol of z_1, $w =$ the rest of z, $x = y = \epsilon$. In the former case we factor $z = uvwxy$ where $u = y = \epsilon$, $v =$ the first a of z, $x =$ the last c of z, $w =$ the rest of z. It is easy to verify that in both cases the pumping conditions for LIN are satisfied.

Thus we have constructed a family of counterexamples (in fact an uncountable family, see [30]) at various levels of the Chomsky hierarchy, each of which satisfies the linear pumping conditions. Now, it is obvious that any language that satisfies the pumping condition for LIN also satisfies the pumping condition for NTBL. Hence, by choosing H appropriately we can have L_H be (properly) at any level of the Chomsky hierarchy while satisfying the pumping condition for NTBL.

For Ogden's lemma

We will present counterexamples, inspired by [26, 13], to show that although Ogden's lemma is an improvement of the pumping lemma, it still does not provide sufficient conditions.

Define $L_G = \{ a^p b^p c^r d^r \mid p, r \geqslant 1 \} \cup \{ a^p b^q c^r d^s \mid 1 \leqslant p < q \text{ and } r, s \geqslant 1 \} \cup$ $\{ a^p b^q c^r d^s \mid 0 < p-q \leqslant r+s \text{ and } p, q, r, s, \geqslant 1 \}$

It is easy to see that L_G is cfl and by [26, lemma 2], L_G is not lcfl. Moreover, as we will argue below, L_G satisfies Ogden's condition of lemma 3.1 with $n = 8$. Consider z in L_G with at least 8 marked positions. $z = w_1 x_1 w_2 x_2 \ldots w_k x_k w_{k+1}$

where the x_i's are all the symbols at the marked positions and w_i's are in Σ^*, $k \geqslant 8$. We only need to argue the case where z is of form $z = a^p b^p c^r d^r$ where p, $r \geqslant 1$. There are 4 possibilities.

(1) $x_2 = a$. If $r > 1$ we choose $u = w_1 x_1 w_2$, $v = x_2$, x = last d, $y = \epsilon$, and w = the rest of string z. Clearly, $d(uvxy) \leqslant 3$. By pumping down we will get a word with the number of a's less than the number of b's. Pumping v and x up will give a word with the number of a's greater than the number of b's and their difference is bounded by the sum of the number of c's and d's. On the other hand, if $r = 1$ then since $d(z) \geqslant 8$, $p \geqslant 3$ and so we can pump a and b. Let $v = x_2$ and x = last b. In this case $d(uvxy) \leqslant 5$. By pumping we will get a word of the same form.

(2) $x_2 = b$. If $p > 1$ we choose $u = w_1 x_1 w_2$, $v = x_2$, $x = y = \epsilon$, and w = the rest of string z. Otherwise, $p = 1$ and thus $r \geqslant 3$ in which case we put $v = x_3$, x = last d, and u, w, and y are defined accordingly. By pumping v and x in both cases, we still get a word in L_G.

(3) $x_2 = c$. This implies that $2r \geqslant 7$ which gives $r \geqslant 4$. Thus, we can pump down c and d. We pick $v = x_2$, x = last d, and u, w, and y are defined accordingly. Pumping v and x will give a word of the same form and thus in L_G.

(4) $x_2 = d$. In this case, we pick v = first c, $x = x_{k-1}$, u, w, and y are defined accordingly. Clearly, pumping will give a word of the same form and thus in L_G.

Hence, L_G satisfies all the three conditions of lemma 3.1 and gives an example of a context-free language that is not linear but satisfies the linear Ogden's conditions.

Let H be a subset of natural numbers, $\Sigma = \{ a, b, c, d \}$ and define $A_H = \{ a^n b^n c^m d^m \mid n, m \text{ in } H \}$, $L_H = A_H \cup \{ a^p b^q c^r d^s \mid p \neq q \text{ or } r \neq s \}$. By [25, theorem 1 and corollary 1] for appropriate H, A_H is properly cfl (i.e., cfl but not lcfl). Moreover, L_H is properly context-sensitive (respectively : recursive, recursively enumerable, not recursively enumerable) if A_H is properly context-sensitive (respectively : recursive, recursively enumerable, not recursively enumerable). We will now prove that L_H satisfies Ogden's conditions of lemma 3.1 with $n = 4$. This will show that there are (uncountably many) languages that satisfy linear Ogden's lemma at various levels of Chomsky hierarchy. Consider z in L_H with at least 4 marked positions. $z = w_1 x_1 w_2 x_2 \ldots w_k x_k w_{k+1}$ where the x_i's are all the symbols at the marked positions and w_i's are in Σ^*, $k \geqslant 4$. If z is in A_H, then we can choose $u = w_1 x_1 w_2$, $v = x_2$ and w = the rest of string of z. By pumping v, we will get a new word with either unequal number of a's and b's or c's and d's. Thus, the three conditions of lemma 3.1 are satisfied.

Now suppose z is not in A_H, i.e., z is of form $z = a^p b^q c^r d^s$ where $p \neq q$. The case for $r \neq s$ can be handled similarly. There are two possible subcases; the first one is when $r \neq s$. We choose $u = w_1 x_1 w_2$, $v = x_2$ and w = the rest of string z. If v is a or b, we will still have $r \neq s$ in any new word obtained by pumping v. On the other hand, if v = c or d, pumping v will still give a word with $p \neq q$. Hence, in this case, the three conditions of lemma 3.1 are satisfied. In the second subcase, we have $r = s$. Without loss of generality, we assume $p > q$. There are four possibilities to be considered in this subcase : (1) when x_2 and x_3 are both a's (2) when $x_2 = a$ and $x_3 = b$ (3) when x_2 and x_3 are both b's and (4) when x_2, x_3 is c or d. We will present the argument for (1) and leave it to the reader to convince himself of the other cases. Consider when x_2 and x_3 are both a's (note $x_1 = a$). If $p - q > 1$ then we put $u = w_1 x_1 w_2$, $v = x_2$ and w = the rest of string z. Obviously, a new word obtained by pumping v will still have $p \neq q$. However, if $p - q = 1$, we choose $u = w_1 x_1 w_2$, $v = x_2 w_3 x_3$ and w = the rest of string z. By pumping v down, we have $p < q$ whereas if v is pumped up we have $p > q$. In both cases, $p \neq q$. We conclude that L_H satisfies all the three conditions of lemma 3.1.

It is obvious that any language that satisfies the linear Ogden's conditions of lemma 3.1 also satisfies the Ogden's condition of theorem 3.1. Hence, by changing H, we obtain counterexamples at various levels of the Chomsky hierarchy, each of which satisfies the Ogden's conditions of theorem 3.1.

CHAPTER 4.

CFL-LCFL NECESSARY CONDITIONS - A COMPARISON

Two famous properties of the class of context-free languages are the (classical) pumping lemma of Bar-Hillel, Perles and Shamir [9] (year 1961) and Parikh's theorem [43] (year 1966). There is no doubt that, beyond the general importance of context-free languages, the popularity of the (classical) pumping lemma stems mainly from the simplicity of its formulation and the ease of its application. In 1968 Ogden [39] proved a considerably stronger pumping lemma for context-free languages; it has since then been named after its discoverer and is nowadays a standard and widely used tool for proving that given languages are not context-free. In 1976 Wise [50] has obtained a strong characterization of context-free languages which has a flavor of pumping. His characterizations provide necessary and sufficient conditions for context-freedom but may become rather unwieldy in applications. In this paper we will not consider such condition because they yield context-freedom.

In 1978 Sokolowski [49] proved another property of context-free languages which was shown to be applicable in some cases where the classical pumping lemma failed. In 1982 Grant [24] proved an extension of the Sokolowski's criterion, Nijholt [38] showed that the latter does not provide a sufficient condition; Bader and Moura [7] proved a generalized Ogden's lemma by introducing the concept of excluded positions (in addition to the distinguished positions in Ogden's lemma). Bader and Moura showed that their lemma is in fact stronger than Ogden's and that it does not provide a sufficient condition for context-freedom. Let us mention in passing that just a slightly weakened version of the main theorem of [7] occurs already in the original paper of Sokolowski [49, lemma 2]. The classes of languages that satisfy the classical pumping conditions and the Ogden's conditions were studied by Horvath and Boasson [30, 13] in 1978.

A pumping lemma for linear context-free languages was mentioned in [11 (sect. v, proposition 6.6), 29 (exercise 6.11)] and an Ogden's lemma for linear languages was proved in chapter 3. In this chapter the generalized Ogden's lemma will be formulated (see also [17]).

We will study the relationships between the various conditions mentioned above. First we present the basic definitions and some introductory results. Three operations on languages are introduced and their relevance to the classes of languages studied in this chapter is indicated. Next we begin the comparison between the general pumping conditions and the linear pumping conditions. In the last section we give several results related to the extended Sokolowski's condition of Grant [24]. We show that the generalized Ogden's lemma [7] is stronger than Grant's condition (implying that the latter condition is not sufficient) and we prove that there are languages that satisfy Grant's condition but not the classical pumping lemma. We end this section by a comparison between the Sokolowski-type conditions and the more standard pumping conditions.

After completion of this research, the authors learned of Horvath's [31] paper which contains some related results, mainly with respect to the pumping conditions (of [9], [39] and [7]) and Sokolowski's conditions (of [49]).

Preliminaries and Elementary Results

In this section we will give some basic definitions, notation and preliminary results. We will formulate various pumping-type conditions for context-free languages [9, 39, 49, 7, 24] and linear context-free languages [11 (proposition 6.6), 29 (exercise 6.11), 17] and will prove some elementary relations between them.

The condition formulated below is a relativized version of the *generalized Ogden's condition* of Bader and Moura [7]. Its special cases include the classical (necessary) conditions of Ogden [39] and Bar-Hillel, Perles and Shamir [9]. Let K be a language.

Generalized Ogden's Condition relative to K (GOC^K) : language $L \subseteq \Sigma^*$ satisfies GOC^K if there exists constant n such that for every z in L and for every marking of positions in z which satisfies $d(z) > n^{e(z)+1}$ there exist u, v, w, x, y in Σ^* such that z = uvwxy and the following conditions hold :

(1) $d(vx) \geqslant 1$ and $e(vx) = 0$

(2) $d(vwx) \leqslant n^{e(vwx)+1}$

(3) for every $i \geqslant 0$, uv^iwx^iy is in $L \cup K$.

The *Generalized Linear Ogden's Condition relative to* K (GOC_l^K) is exactly like GOC^K except that the assumption $d(z) > n^{e(z)+1}$ is replaced by the assumption $d(z) > n(e(z)+1)$ and condition (2) is replaced by the condition : $d(uvxy) \leqslant n(e(uvxy)+1)$. When $K = \varnothing$, GOC^K (respectively GOC_l^K) gives *Generalized Ogden's Condition* denoted by GOC (respectively, *Generalized Linear Ogden's Condition* denoted by GOC_l). Special, well-known, cases of these conditions are obtained by restricting the functions d and e. Thus GOC (respectively GOC_l) when restricted by requiring $e(z) = 0$ for all z, gives the well-known *Ogden's Condition* [39] abbreviated by OC (respectively *Linear Ogden's Condition*, abbreviated by OC_l). Further restriction of OC (respectively OC_l) requiring $d(z) = |z|$ for all z, gives the *Pumping Condition* [9], denoted by PC (respectively, *Linear Pumping Condition*, denoted by PC_l).

To formulate the Sokolowski-type conditions [49, 24] we need some further concepts. Let Σ be an alphabet. A binary relation on Σ^* is said to be *unbounded* if for every m there exist x, y in Σ^* such that $|x|, |y| \geqslant m$ and $R(x,y)$. For x, y in Σ^* we will write $x < y$ if x is obtained from y by deleting at least one symbol. If all the deleted symbols are from among the last (first) m symbols of y we write $x <^m y$ ($x \;{}^{m}\!\!< y$). Now we define

$$(\hat{x},\hat{y}) <_m (x,y) \leftrightarrow [\,\hat{x} < x \;\&\; \hat{y} = y\,] \vee [\,\hat{x} = x \;\&\; \hat{y} < y\,] \vee [\,\hat{x} <^m x \;\&\; \hat{y} \;{}^{m}\!\!< y\,].$$

Extended Sokolowski's Condition relative to K (ESC^K) : language $L \subseteq \Sigma^*$ satisfies ESC^K if for every u_1,u_2,u_3 in Σ^* and every unbounded relation R on Σ^*, if $\{\, u_1xu_2yu_3 \mid R(x,y) \,\} \subseteq L$ then

$$\exists m \,\forall x, y\, [\; |x|, |y| > m \;\&\; R(x,y) \rightarrow$$
$$\exists \hat{x}, \hat{y}\, [\, (\hat{x},\hat{y}) <_m (x,y) \;\&\; u_1\hat{x}u_2\hat{y}u_3 \in L \cup K \,]\,].$$

When R is an equality relation on a subset Σ_R^* where $\Sigma_R \subseteq \Sigma$ with $|\Sigma_R| \geqslant 2$ then ESC^K gives the *Sokolowski's condition relative to* K (SC^K). When $K = \varnothing$, ESC^K and SC^K reduce respectively to *Extended Sokolowski's Condition* (ESC) of [24] and *Sokolowski's Condition* (SC) of [49].

The name of a class of languages which satisfy (one of) these conditions is obtained from the name of the condition by replacing the letter C by L; thus GOL_l is the class of languages that satisfy GOC_l.

Lemma 4.1. Let CON be any of the conditions GOC, GOC_l, OC, OC_l, PC, PC_l and suppose that K and L satisfy CON^L and CON^K respectively. Then $L \cup K$ satisfies CON.

Proof. Suppose that K and L satisfy GOC^L and GOC^K with respective constants n_1 and n_2. Let $n = \max\{n_1, n_2\}$. We will show that $L \cup K$ satisfies GOC with constant n. Consider z in $L \cup K$ with marking that satisfies $d(z) > n^{e(z)+1}$. If z is in K then since $d(z) > n_1^{e(z)+1}$ and since K satisfies GOC^L, there exists a factorization $z = uvwxy$ such that (1) $d(vx) \geqslant 1$ and $e(vx) = 0$, (2) $d(vwx) \leqslant n_1^{e(vwx)+1} \leqslant n^{e(vwx)+1}$ and (3) for every $i \geqslant 0$, uv^iwx^iy is in $L \cup K$. On the other hand, if z is in L then since L satisfies GOC^K with constant n_2 and since $d(z) > n_2^{e(z)+1}$, z has a factorization $z = uvwxy$ with (1) $d(vx) \geqslant 1$ and $e(vx) = 0$, (2) $d(vwx) \leqslant n_2^{e(vwx)+1} \leqslant n^{e(vwx)+1}$ and (3) for every $i \geqslant 0$, uv^iwx^iy is in $L \cup K$. Hence $L \cup K$ satisfies GOC. The proofs of other cases are very similar. □

Remark. Lemma 4.1 is useful (for example) in a situation where $L = L_1 \cup L_2$, L_2 is lcfl and we want to prove that L satisfies GOC_l; then it is sufficient to consider words in L_1.

Lemma 4.2. If K satisfies ESC^L (SC^L) and L satisfies ESC^K (SC^K) then $L \cup K$ satisfies ESC (SC).

Proof. Let R be an unbounded relation on Σ^* and u_1,u_2,u_3 in Σ^* such that $\{ u_1xu_2yu_3 \mid R(x,y) \} \subseteq L \cup K$. Define binary relations $R_L(x,y) \leftrightarrow [\, R(x,y)$ and $u_1xu_2yu_3 \in L\,]$ and $R_K(x,y) \leftrightarrow [\, R(x,y)$ and $u_1xu_2yu_3 \in K\,]$. If R_L is not unbounded then there exists a constant r such that for every x, y in Σ^* $[\, R_L(x,y) \rightarrow (\,|x| < r$ or $|y| < r\,)\,]$. In that event put $m_L = r$. Otherwise, R_L is unbounded and $\{ u_1xu_2yu_3 \mid R_L(x,y) \} \subseteq L$. Since L satisfies ESC^K, there exists a constant q such that for every x, y in Σ^*

$$|x|,\ |y| > q \ \&\ R_L(x,y) \rightarrow \exists \hat{x}, \hat{y}\ [\, (\hat{x},\hat{y}) <_q (x,y) \ \&\ u_1\hat{x}u_2\hat{y}u_3 \in L \cup K\,] \qquad *)$$

In this case put $m_L = q$. Define m_K in a similar manner and set $m = \max(m_L, m_K)$. Now let x, y in Σ^* be such that $|x|,\ |y| > m$ and $R(x,y)$; since $u_1xu_2yu_3 \in L \cup K$, we have either $R_L(x,y)$ or $R_K(x,y)$. If $R_L(x,y)$ then R_L must be unbounded and hence, by (*), there exist $\hat{x}$ and $\hat{y}$ such that $(\hat{x},\hat{y}) <_q (x,y)$ and $u_1\hat{x}u_2\hat{y}u_3$ is in $L \cup K$. Since $q \leqslant m$ we also have $(\hat{x},\hat{y}) <_m (x,y)$ and hence $L \cup K$ satisfies ESC with constant m. The case when $R_K(x,y)$ holds (rather than $R_L(x,y)$) is argued similarly.

The proof for SC is similar. □

The following corollaries follow from lemmas 4.1 and 4.2 and the observation that if a language satisfies any of the conditions in these lemmas it also satisfies the relativized version of that condition.

Corollary 1. The classes of languages GOL, GOL_l, OL, OL_l, PL, PL_l, ESL, SL are closed under union.

Corollary 2. Let CON be any of the conditions defined above and let K and L be languages over disjoint alphabets. Then K $\cup$ L satisfies CON iff both K and L satisfy CON.

We end this section with a result which will be used later.

Lemma 4.3. PL, OL and GOL are closed under concatenation.

Proof. Suppose K and L satisfy GOC with constants m and n respectively. We want to show that K·L satisfies GOC with a constant m + n. Consider $z = z_1z_2$ where z_1 is in K and z_2 is in L such that $d(z) > (m+n)^{e(z)+1}$. Then either $d(z_1) > m^{e(z_1)+1}$ or $d(z_2) > n^{e(z_2)+1}$ for otherwise $d(z) = d(z_1) + d(z_2) \leqslant m^{e(z_1)+1} + n^{e(z_2)+1} \leqslant (m+n)^{e(z_1)+e(z_2)+1} = (m+n)^{e(z)+1}$ contradicting our assumption. Without loss of generality, assume $d(z_1) > m^{e(z_1)+1}$. Since K satisfies GOC with constant m, there exist u, v, w, x and y where z_1 = uvwxy such that (1) $d(vx) \geqslant 1$ and $e(vx) = 0$ (2) $d(vwx) \leqslant m^{e(vwx)+1}$ (3) for every $i \geqslant 0$, uv^iwx^iy is in K. Put $y_1 = yz_2$. We have $z = uvwxy_1$ such that condition (1) holds ; moreover (2) $d(vwx) \leqslant m^{e(vwx)+1} < (m+n)^{e(vwx)+1}$ and (3) for every $i \geqslant 0$, $uv^iwx^iy_1$ is in K·L. Hence, K·L is in GOL. The proofs for other cases are similar. □

Language Operations Related to Pumping

We will now define three operations on languages and then investigate the relationship between these operations and the pumping conditions defined in previous sections. The results will provide us with a systematic way of proving correctness of inclusion diagrams (like those in figures 9 and 10) by allowing a quick way of constructing appropriate counterexamples.

Let Σ be an alphabet which does not include f or g and let $L \subseteq \Sigma^*$. We define three operators :

<u>a-operation</u> :

$$L^a = L \cdot \{ f^n g^n \mid n \geqslant 1 \} \cup \Sigma^* \cdot \{ f^n g^m \mid n, m \geqslant 1, n \neq m \} \cup \Sigma^*$$

<u>r-operation</u> :

$$L^r = L \cdot \{ f^n g^n \mid n \geqslant 1 \} \cup \Sigma^* \cdot \{ f^n g^m \mid n \neq m \}$$

<u>s-operation</u> :

$$L^s = \{ f^n z g^n \mid z \text{ in } L, n \geqslant 1 \} \cup \Sigma^*$$

The first theorem summarizes the "universal" effect of these operations, i.e., without making any assumptions about the argument languages. For notational convenience we will denote by $L^i_{(j)}$ the "j^{th} part" of L^i where i is a, r or s. For example, $L^a_{(3)} = \Sigma^*$ and $L^r_{(2)} = \Sigma^* \cdot \{ f^n g^m \mid n \neq m \}$.

Theorem 4.1. For any language L, (i) L^a is in PL (ii) L^r is in both OL and OL_l (iii) L^s is in PL_l.

Proof. It is easy to show that L^s is in PL_l. We will omit the proof. To see that L^a is in PL first note that $L^a_{(2)} \cup L^a_{(3)}$ is cfl and hence in PL. By lemma 4.1 it suffices to show that $L^a_{(1)}$ satisfies PC relative to $L^a_{(2)} \cup L^a_{(3)}$. Take n = 2 as the PC constant and consider z in $L^a_{(1)}$ with $|z| > 2$. Choose v = the rightmost f, w = ϵ and x = the leftmost g. Define u and y accordingly. Clearly z = uvwxy, $|vx| \geqslant 1$, $|vwx| \leqslant 2$ and uv^iwx^iy in $L^a_{(1)} \cup L^a_{(3)}$ for all $i \geqslant 0$. Hence $L^a_{(1)}$ satisfies PC relative to

$L^a_{(2)} \cup L^a_{(3)}$ and so is in PL.

To show that L^r is in both OL and OL_l we take the constant 2. Since $L^r_{(2)}$ is lcfl, it satisfies both OC and OC_l. By lemma 4.1, it suffices to show that $L^r_{(1)}$ satisfies OC and OC_l relative to $L^r_{(2)}$. Consider $z = z_1 f^n g^n$ where z_1 is in L, $n \geqslant 1$ and such that $d(z) > 2$. We will first show that $L^r_{(1)}$ satisfies OC relative to $L^r_{(2)}$. If the substring $f^n g^n$ contains distinguished positions we let $x = y = \epsilon$, v = the rightmost distinguished position and define u and w accordingly. Otherwise, since $d(z) > 2$ there must be at least 3 distinguished positions in z_1 and we put v = the rightmost distinguished position of z_1, x = the first f and u, w, y are defined accordingly. In both cases, $d(vx) \geqslant 1$, $d(vwx) \leqslant 2$ and uv^iwx^iy is in $L^r_{(1)} \cup L^r_{(2)}$ for all $i \geqslant 0$. Thus, $L^r_{(1)}$ satisfies OC relative to $L^r_{(2)}$. As for OC_l, if the substring $f^n g^n$ contains any distinguished position then let $u = v = \epsilon$, x = the rightmost distinguished position and w and y are defined accordingly. Otherwise, take v = the leftmost distinguished position, x = the last g and u, w, y are defined accordingly. Clearly, in both cases $d(vx) \geqslant 1$, $d(uvxy) \leqslant 2$ and for every $i \geqslant 0$, uv^iwx^iy is in $L^r_{(1)} \cup L^r_{(2)}$. We conclude that $L^r_{(1)}$ satisfies OC and OC_l relative to $L^r_{(2)}$. By lemma 4.1, L^r satisfies OC and OC_l. □

We proceed with a technical lemma. For notational convenience, if a word z is factorized as $z = uvwxy$ then by $z^{(i)}$ we denote the word uv^iwx^iy.

Lemma 4.4. Let z be a string of the form $z_1 fg$ (fz_1g) where z_1 is in Σ^* and f, g are symbols not in Σ. For any factorization $z = uvwxy$ for which (1) $|vx| \geqslant 1$ and, (2) v and x are within z_1(if nonempty), we can define an induced factorization of z_1, $z_1 = u_1vw_1xy_1$ such that (3) $z^{(i)} = z_1^{(i)}fg$ ($fz_1^{(i)}g$) for every $i \geqslant 0$ and (4) $|u_1vxy_1| \leqslant |uvxy|$ and $|vw_1x| \leqslant |vwx|$.

Proof. Let $z = z_1fg$ as described in the lemma. Then put $u_1 = u$ and define w_1 and y_1 as follows. If $x \neq \epsilon$ then $w_1 = w$ and y_1 is obtained from y by deleting f and g. Suppose $x = \epsilon$. Then if $y = \epsilon$ put $y_1 = \epsilon$ and w_1 is obtained from w by deleting f and g ; if $y = g$ then put $y_1 = \epsilon$ and w_1 is obtained from w by deleting f ; if $y = y'fg$ then put $y_1 = y'$ and $w_1 = w$. Obviously, $z^{(i)} = z_1^{(i)}fg$ and since u_1, w_1 and y_1 are either the same as u, w and y or shorter, we have $|u_1vxy_1| \leqslant |uvxy|$ and $|vw_1x| \leqslant |vwx|$. The case for $z = fz_1g$ can be proved similarly. □

The next result shows how the various classes of languages defined in previous section are preserved under the a - and r - operations.

Theorem 4.2.

(i) L is in OL (OL_l) iff L^a is in OL (OL_l).

(ii) Let e be in { a, r }. L is in GOL (respectively GOL_l, ESL and SL) iff L^e is in GOL (respectively GOL_l, ESL and SL).

Proof. (i) We will only argue the OL_l–case. For the OL–case the proof is similar. Suppose that L satisfies OC_l with constant n. To show that L^a is in OL_l, by lemma 4.1, it suffices to show that $L^a_{(1)}$ satisfies OC_l relative to $L^a_{(2)} \cup L^a_{(3)}$ with constant $n + 2$. Take $z = z_1f^mg^m$ from $L^a_{(1)}$ such that $d(z) > n + 2$. If there is some distinguished position in the substring f^mg^m and $m > 1$, take $x =$ the rightmost such position, $u = v = \epsilon$, w and y are defined accordingly. Clearly, $d(vx) \geqslant 1$, uv^iwx^iy is in $L^a_{(1)} \cup L^a_{(2)}$ for every $i \geqslant 0$ and $d(uvxy) = 1 \leqslant n$. Otherwise, we have $d(z_1) =$

$d(z) > n + 2 > n$ when $d(f^m g^m) = 0$ and $d(z_1) = d(z) - d(fg) > n$ when $m = 1$. Since z_1 is in L which satisfies OC_l with constant n, we can write $z_1 = uvwxy$ such that (1) $d(vx) \geqslant 1$, (2) $d(uvxy) \leqslant n$ and (3) for every $i \geqslant 0$, uv^iwx^iy is in L. Thus $z = z_1 f^m g^m = uvwxy'$ where u, v, w, x are as above and $y' = yf^m g^m$. Hence, condition (1) holds; moreover, (2) $d(uvxy') \leqslant n + 2$ and (3) for every $i \geqslant 0$, uv^iwx^iy' is in $L^a_{(1)}$. In both cases, we conclude that $L^a_{(1)}$ satisfies OC_l relative to $L^a_{(2)} \cup L^a_{(3)}$. Note that for OL-case, the proof holds by corollary 1 and lemma 4.3.

Conversely, suppose L^a satisfies OC_l with constant n. We want to show that L is in OL_l. If L does not satisfy OC_l then there exists z_1 in L such that $d(z_1) > n$ and for every u, v, w, x and y for which $z_1 = uvwxy$, $d(vx) \geqslant 1$ and $d(uvxy) \leqslant n$ there exists $i \geqslant 0$ such that $z_1^{(i)}$ is not in L. Consider $z = z_1 fg$ where f and g are not distinguished. Then $d(z) = d(z_1) > n$. Since L^a satisfies OC_l with constant n, we may write $z = uvwxy$ such that (1) $d(vx) \geqslant 1$, (2) $d(uvxy) \leqslant n$ and (3) for every $i \geqslant 0$, $z^{(i)}$ is in L^a. From (3) we can easily infer that neither v nor x can contain f or g. Hence it follows that v and x must be within z_1. By lemma 4.4 and since f and g are not distinguished positions, the above factorization of z induces a factorization of $z_1 = u_1 v w_1 x y_1$ such that (i) $d(vx) \geqslant 1$ (ii) $d(u_1 vxy_1) = d(uvxy) \leqslant n$. Thus there exists $i \geqslant 0$ such that $z_1^{(i)}$ is not in L. But since $z^{(i)} = z_1^{(i)} fg$ this implies that $z^{(i)}$ is not in L^a for some i which contradicts condition (3) above. Hence, L^a satisfies OC_l.

(ii) We will show that L is in GOL iff L^r is in GOL. If L is in GOL then by corollary 1 and lemma 4.3, L^r is in GOL. Conversely, suppose L^r satisfies GOC with constant n. If L does not satisfy GOC then there exists z_1 in L such that $d(z_1) > n^{3(e(z_1)+1)}$ and for every u, v, w, x and y such that $z_1 = uvwxy$ and if $d(vx) \geqslant 1$, $e(vx) = 0$ and $d(vwx) \leqslant n^{3(e(vwx)+1)}$ then $z_1^{(i)} = uv^iwx^iy$ is not in L for some $i \geqslant 0$. Now consider $z = z_1fg$ in L^r where f and g are excluded but not distinguished. We have $d(z) = d(z_1) > n^{3(e(z_1)+1)} = n^{3(e(z)-1)} \geqslant n^{e(z)+1}$. Since L^r satisfies GOC, we may write $z = uvwxy$ such that (i) $d(vx) \geqslant 1$, $e(vx) = 0$ (ii) $d(vwx) \leqslant n^{e(vwx)+1}$ (iii) for every $i \geqslant 0$, $z^{(i)}$ is in L^r. By condition (i), v and x must be within z_1 and thus by lemma 4.4 we can obtain the induced factorization of z_1 as $u_1vw_1xy_1$ such that (1) $d(vx) \geqslant 1$, $e(vx) = 0$ (2) $d(vw_1x) = d(vwx) \leqslant n^{e(vwx)+1} \leqslant n^{e(vw_1x)+3} \leqslant n^{3(e(vw_1x)+1)}$. Thus, there exists $i \geqslant 0$ such that $z_1^{(i)}$ is not in L. But by lemma 4.4 $z^{(i)} = z_1^{(i)}fg$ which implies that $z^{(i)}$ is not in L^r for some i contradicting condition (iii) above. Hence, L satisfies GOC.

Now we will show that L is in GOL_l iff L^r is in GOL. Suppose that L satisfies GOC_l with constant n. By lemma 4.1, it suffices to show that $L^r_{(1)}$ satisfies GOC_l relative to $L^r_{(2)}$ with constant n. Take $z = z_1f^mg^m$ from $L^r_{(1)}$ such that $d(z) > n(e(z)+1)$. If there is some $\bar{d}\bar{e}\bar{p}$ in the substring f^mg^m, take x = the rightmost such position, $u = v = \epsilon$, w and y are defined accordingly. Clearly, $d(vx) = 1$ and $e(vx)=0$, uv^iwx^iy is in L^r for every $i \geqslant 0$ and $d(uvxy) \leqslant 1 + e(y) \leqslant n(e(uvxy)+1)$. Otherwise, we have $d(f^mg^m) \leqslant e(f^mg^m)$ and so $d(z_1) = d(z) - d(f^mg^m) > n(e(z)+1) - e(f^mg^m) \geqslant n(e(z)+1-e(f^mg^m)) = n(e(z_1)+1)$. Since z_1 is in L which satisfies GOC_l with constant n, we can write $z_1 = uvwxy$ such that (1) $d(vx) \geqslant 1$, $e(vx) = 0$, (2) $d(uvxy) \leqslant n(e(uvxy)+1)$ and (3) for every $i \geqslant 0$, uv^iwx^iy is in L.

We use this factorization of z_1 to obtain a factorization of z. Thus $z = z_1 f^m g^m = uvwxy'$ where u, v, w, x are as above and $y' = yf^m g^m$. Hence, conditions (1) and (3) hold and moreover (2) $d(uvxy') \leqslant n(e(uvxy)+1) + e(f^m g^m) \leqslant n(e(uvxy') + 1)$. In both cases, we conclude that $L^r_{(1)}$ satisfies GOC_l relative to $L^r_{(2)}$. Conversely, if L^r is in GOL_l we can show that L is in GOL_l similarly to the GOC-case.

The proofs for the a-operation with respect to GOC and GOC_l are similar and so we leave them out.

We will now prove that L is in ESL iff L^a is in ESL. It is easy to show that if L^a is in ESL then L is in ESL and so we omit this part. Conversely, suppose L is in ESL and we want to show that L^a is in ESL. Let $\Sigma_a = \Sigma \cup \{f, g\}$. Since $L^a_{(2)} \cup L^a_{(3)}$ is cfl, by lemma 4.2, it suffices to show that $L^a_{(1)}$ satisfies ESC relative to $L^a_{(2)} \cup L^a_{(3)}$. Let R be an unbounded binary relation over Σ_a^* and u_1, u_2, u_3 in Σ_a^* satisfy $\{ u_1 x u_2 y u_3 \mid R(x,y) \} \subseteq L^a_{(1)}$ (*). We want to show that

$$\exists m \forall x, y [\ |x|, |y| > m \ \&\ R(x,y) \rightarrow \exists \hat{x}, \hat{y} [(\hat{x}, \hat{y}) <_m (x,y) \ \&\ u_1 \hat{x} u_2 \hat{y} u_3 \in L^a]] \qquad (**)$$

If $u_3 = \epsilon$ then choose m = 2. Let x, y be such that $|x|, |y| > 2$ and R(x,y). By (*), $u_1 x u_2 y$ is of the form $zf^n g^n$ for some $n \geqslant 1$ with z in L. $|y| > 2$ implies that either y contains at least two g's or y contains exactly one f and one g. We can then obtain $\hat{y}$ from y by deleting a g in the former case and fg in the latter. In both cases $u_1 x u_2 \hat{y} \in L^a$ and thus (**) holds with m = 2. Otherwise, when $u_3 \neq \epsilon$ there are two cases to be considered.

Case 1 : u_3 has equal number of f's and g's. Then u_3 must be of the form $v_3 f^n g^n$ for some v_3 in Σ^* and $n \geqslant 1$. In this case R must in fact be over Σ^* and we have $\{ u_1xu_2yv_3 \mid R(x,y) \} \subseteq L$. Since R is unbounded and since L satisfies ESC, there exists a constant m such that for every x, y in Σ_a^*

$$|x|, |y| > m \ \& \ R(x,y) \rightarrow \exists \hat{x}, \hat{y} \, [(\hat{x}, \hat{y}) <_m (x, y) \ \& \ u_1\hat{x}u_2\hat{y}v_3 \in L] \ .$$

Clearly, $u_1\hat{x}u_2\hat{y}v_3 \in L$ implies that $u_1\hat{x}u_2\hat{y}u_3 \in L^a_{(1)}$ and so (**) holds true.

Case 2 : u_3 has unequal number of f's and g's. Obviously the number of f's in u_3 must be less than the number of g's. If u_3 contains f's put m = 1. Then for any x, y with $|x|, |y| > 1$ and R(x,y) y must contain f's and we obtain $\hat{y}$ by deleting one f from y. Clearly, $u_1xu_2\hat{y}u_3 \in L^a_{(2)}$ and hence (**) holds for m = 1. Otherwise, u_3 is of the form g^k for some $k \geqslant 1$. If $k > 1$ choose m = 1. Then for x, y such that $|x|, |y| > 1$ and R(x,y) we define $\hat{y}$ by deleting one f or g from y. Thus $u_1xu_2\hat{y}g^k \in L^a_{(2)}$ and (**) holds. On the other hand if k = 1, i.e., $u_3 = g$, then there are two possibilities for u_2. If u_2 contains f or g then we argue exactly as when k > 1. Otherwise, u_2 does not contain f or g and we define a binary relation over Σ^* : $R'(x,y) \leftrightarrow \exists i \geqslant 0 \, [R(x,yf^{i+1}g^i)]$. If R' is not unbounded then there exists a constant r such that for every x, y in $\Sigma^* \, [R'(x, y) \rightarrow (|x| < r \text{ or } |y| < r)]$. In this case put m = r + 1. Otherwise, R' is unbounded. Then $\{ u_1xu_2y \mid R'(x, y) \} \subseteq L$ because $R'(x, y)$ implies $R(x, yf^{i+1}g^i)$ for some $i \geqslant 0$ and so by (*), we have $u_1xu_2yf^{i+1}g^ig \in L^a_{(1)}$ which implies that $u_1xu_2y \in L$. Since L satisfies ESC, let s be the corresponding constant. In this case put m = s + 1. Now let x, y be in Σ_a^* such that $|x|, |y| > m$ and R(x,y). By (*) $u_1xu_2yu_3 \in L^a_{(1)}$ and y must be of the form $y_1f^{i+1}g^i$ for some $i \geqslant 0$ with y_1 in Σ^*. Clearly $R'(x, y_1)$ holds. If R' is not unbounded, then $R'(x, y_1)$ and $|x| > m = r + 1$ implies that $|y_1| < r$. Thus $|f^{i+1}g^i| =$

$|y| - |y_1| > m - r = 1$ and so y has at least two f's. We can then obtain $\hat{y}$ from y by deleting one f and get $u_1xu_2\hat{y}u_3 \in L^a_{(2)}$. Otherwise, R' is unbounded. If $i \geqslant 1$ we obtain $\hat{y}$ by deleting one f from y and so $u_1xu_2\hat{y}u_3 \in L^a_{(2)}$. Otherwise, $y = y_1f$ and so $|x|, |y| > m = s + 1$ implies that $|x|, |y_1| > s$. Thus, we have $|x|, |y_1| > s$ and $R'(x,y_1)$. Since L satisfies ESC with constant s, there exist $\hat{x}$, $\hat{y}_1$ such that $(\hat{x},\hat{y}_1) <_s (x,y_1)$ and $u_1\hat{x}u_2\hat{y}_1 \in L$. This implies that $u_1\hat{x}u_2\hat{y}_1fg \in L^a_{(1)}$ i.e., $u_1\hat{x}u_2\hat{y}u_3 \in L^a$ where $\hat{y} = \hat{y}_1f$. Since $s < m$, clearly $(\hat{x}, \hat{y}) <_s (x, y)$ implies $(\hat{x}, \hat{y}) <_m (x,y)$ and thus (**) holds. Hence we conclude that L^a is in ESL and the proof is complete . □

To explain the intuitive idea of our next result let A and B be classes of languages, $A \subset B$, and let f be a language-theoretic operation. Then our result is a kind of "downward bridge theorem" in the sense that :

$$L \text{ not in } B \Rightarrow f(L) \text{ not in } A .$$

This becomes useful when f(L) is in B because then we have a separation result : f(L) is in B but not in A . We now present our last theorem of this section.

Theorem 4.3. If L is not in PL (PL_l) then L^s is not in OL (OL_l).

Proof. We will first show that L^s is not in OL. Suppose L^s satisfies OC with constant n. Since L does not satisfy PC we can choose z_1 in L such that $|z_1| > n + 1$ and for every u, v, w, x and y such that $z_1 = uvwxy$, if $|vx| \geqslant 1$ and $|vwx| \leqslant n + 1$ then $z_1^{(i)} = uv^iwx^iy$ is not in L for some $i \geqslant 0$. Now consider $z = fz_1g$ in L^s where all positions of z_1 are distinguished but f and g are not distinguished. We have $d(z) = |z_1| > n + 1 > n$. Since L^s satisfies OC, we may write $z = uvwxy$

such that (i) $d(vx) \geqslant 1$ (ii) $d(vwx) \leqslant n$ (iii) for every $i \geqslant 0$, $z^{(i)}$ is in L^s. By conditions (i) and (iii), v and x must be within z_1 and thus by lemma 4.4 we can obtain the induced factorization of z_1 as $u_1vw_1xy_1$ such that $|vx| = d(vx) \geqslant 1$ and $|vw_1x| \leqslant |vwx| \leqslant d(vwx) + 1 \leqslant n + 1$. Since L does not satisfy PC there exists $i \geqslant 0$ such that $z_1^{(i)}$ is not in L. By lemma 4.4, we have $z^{(i)} = fz_1^{(i)}g$ which implies that $z^{(i)}$ is not in L^s for some $i \geqslant 0$. This contradicts condition (iii) above. Hence, L^s does not satisfy OC. The OL_l -case is similar. □

Comparison between Context-free Pumping Conditions (PC's) and Linear Context-free Pumping Conditions (PC_l's)

Bader and Moura [7] have shown that CFL $\subsetneqq$ GOL $\subsetneqq$ OL. It is also well-known that OL $\subsetneqq$ PL. These results together establish the correctness of the inclusion diagram of figure 6 where the languages K_i will be defined later.

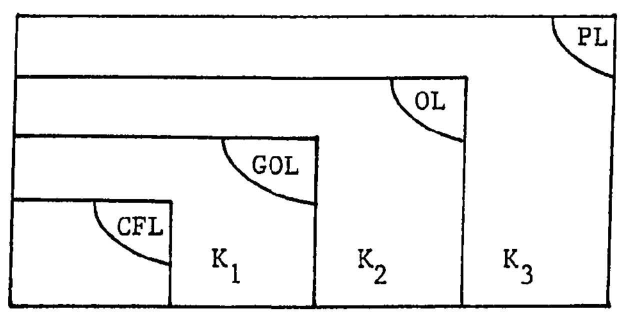

Figure 6 : Hierarchy of languages satisfying various pumping conditions

With respect to LIN we prove the generalized Ogden's lemma.

Theorem 4.4 (The generalized Ogden's lemma). LIN $\subseteq$ GOL_l .

Proof. Let $G = (N, T, P, S)$ be a lcfg for L with k nonterminals and put $n = (k + 1)p$. Let z in L satisfy $d(z) > (e(z) + 1)n$ and consider a derivation tree t for a derivation $S \overset{*}{\underset{G}{\Rightarrow}} z$. Let C be a path in t with maximum number of branch nodes. Since $d(z) > (e(z) + 1)n \geq (e(z) + 1)n - p + 1 = ((k + 1)(e(z) + 1) - 1)p + 1$, claim 3.1, chapter 3, implies that C has at least $(k + 1)(e(z) + 1)$ branch nodes. We divide the uppermost part of C into $e(z) + 1$ subpaths, each containing $k + 1$ branch nodes. Since there are k nonterminals, each subpath i must have two branch nodes with the same label, say A_i, $i = 1, \ldots, e(z) + 1$. Thus, there exist two strings of terminals v_i and x_i (called a pumping pair) such that $A_i \underset{G}{\Rightarrow} \alpha_i B_i \beta_i \overset{*}{\underset{G}{\Rightarrow}} v_i A_i x_i$ where A_i and B_i are in N. Since the upper A_i is a branch node, $v_i x_i$ must contain at least one distinguished position. Starting from the top of path C, we proceed through the subpaths until we find a pumping pair that contains no excluded positions. Such a pair surely exists because there are $e(z) + 1$ distinct pairs but only $e(z)$ excluded positions. Call this pair v, x which now obviously satisfies (1) and (3) of GOC_l.

It remains to show that condition (2) holds as well. Suppose that v, x are in the (g+1)-st subpath from the top of C and let A be the corresponding nonterminal (i. e., $A = A_{g+1}$). For each subpath above this one, the pumping pair contains at least one excluded position. Thus $e(uvxy) \geq g$. By the definition of g, the subpath (along the path C) from the root of t to the direct ancestor of the lower A contains at most $(k + 1)(g + 1) - 1$ branch nodes. Hence, by claim 3.1 $d(uvxy) \leq ((k + 1)(g + 1) - 1)p + 1 \leq (k + 1)(e(uvxy) + 1)p - p + 1 \leq (e(uvxy) + 1)n$. This completes the proof of the theorem. □

Remark. With some small modifications in the above proof, we may obtain a slightly stronger version of the generalized Ogden's lemma. Specifically, in condition (1) we can have either each of u, v, w or each of w, x, y contain at least one distinguished position while v and x having no excluded positions.

Corollary 3. (Linear Ogden's lemma). For any lcfl L, L satisfies OC_l.

Proof. This is an immediate consequence of theorem 4.4. (see also lemma 3.1, chapter 3) □

Like in the case of generalized Ogden's lemma for context-free languages, see [7], the generalized Ogden's lemma for LIN is stronger than Ogden's lemma for LIN. In fact, relationships similar to those depicted in figure 6 hold for the respective linear conditions, see figure 7. The languages M_i will be defined below.

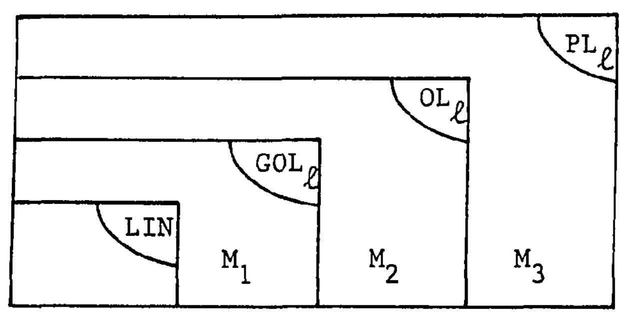

Figure 7 : Hierarchy of languages satisfying various linear pumping conditions

We have seen two pumping hierarchies, the (general) context-free : CFL $\subsetneq$ GOL $\subsetneq$ OL $\subsetneq$ PL and the linear context-free : LIN $\subsetneq$ GOL_l $\subsetneq$ OL_l $\subsetneq$ PL_l , see figures 6 and 7. We will now study the relationships between the two hierarchies, i.e., we will put the inclusion diagrams of figures 6 and 7 into one comprehensive picture. Generally speaking, we will see that the linear and the general pumping conditions are orthogonal as suggested by figure 8.

The tools used to prove the correctness of the diagrams will be the corollaries of lemma 4.1, lemma 4.2 and the results of the previous section. In figure 8 there are twenty "areas of interest" numbered by integers 1 to 20. L_i denotes a language that will fit in the area i.

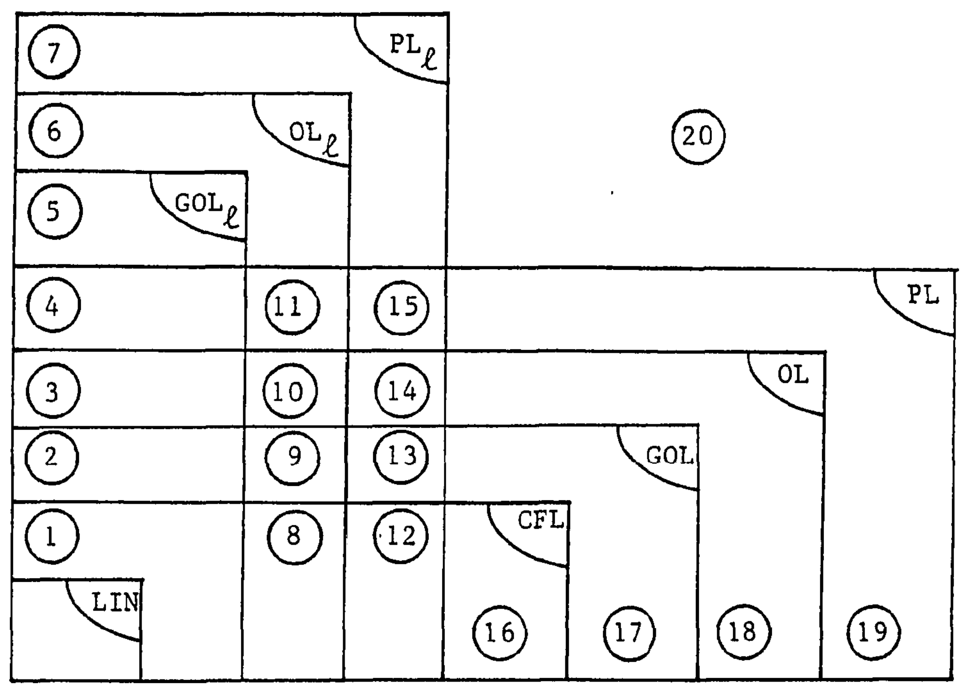

Figure 8 : Comparison between context-free and linear pumping conditions

Define M_1 (in figure 7) to be the language $\{ a^p b^q c^r d^s \mid 1 \leqslant p < q$ and $r, s \geqslant 1 \} \cup \{ a^p b^p c^r d^r \mid p, r \geqslant 1\} \cup \{ a^p b^q c^r d^s \mid 0 < p-q \leqslant r+s$ and $p, q, r, s \geqslant 1 \}$. By the lemma of Greibach [26, lemma 2] M_1 is not lcfl and so it remains to show that M_1 satisfies GOC_l. Since the first and the last "parts" of M_1 are lcfl's, by lemma 4.1 it suffices to show that the second part satisfies GOC_l relative to the other two. Consider $z = a^p b^p c^r d^r$ where $p, r \geqslant 1$ with a marking that satisfies $d(z) > 4(e(z) + 1)$. Then z has at least 5 distinguished positions which are not excluded. Since $p = r = 1$ is impossible there are three cases to be considered. (i) $p = 1$ (ii) $r = 1$ (iii) $p > 1$ and $r > 1$. We will argue the third case and leave the other two to the reader. Let s and t be the leftmost and rightmost distinguished but not excluded positions of z respectively. There are four subcases of which only one will be discussed in some detail. This case is when $s = a$ (i.e., the symbol in position s is a). If $t \neq a$ then let $v = s$, $x = t$ and define u, w and y accordingly. Clearly, $d(uvxy) \leqslant e(uvxy) + 2 < 4(e(uv\,xy) + 1)$ and pumping will either produce a word of the same form (when $t = b$) or words with number of a's < b's or words with number of a's > b's with the difference bounded by the number of c's plus the number of d's (when $t = c$ or d); in all these cases the resulting words are in M_1. Otherwise, if $t = a$ then not all b's, c's and d's can be excluded (if they were, we would have $e(z) = e(a^p) + p + 2r \geqslant p + 2r$ implying $4(e(z) + 1) \geqslant 4(p + 2r + 1) > 2p + 2r \geqslant d(z)$ and contradicting our assumption on z) and we let x to be the rightmost such nonexcluded position. Define $v = s$ and u, w and y accordingly. Then $d(u) \leqslant e(u)$, $d(y) \leqslant e(y)$, $d(x) = e(x) = 0$, $d(v) = 1$, $e(v) = 0$ and so $d(uvxy) \leqslant 1 + e(uvxy) < 4(e(uv\,xy) + 1)$. Again pumping will produce words in M_1. The other subcases, when s is b or c or d, are argued similarly.

Let L_1 be the language M_1. Since L_1 is cfl and was shown to be in GOL_l −LIN it exactly fits area 1. In [7], the language $K_1 = \{ z \text{ in } \{a,b\}^* \mid z = (ab)^q \Rightarrow q$ prime $\}$ was proved to be located as shown in figure 6. Let L_2 be the language K_1. To show that L_2 fits area 2 we need to show that L_2 satisfies GOC_l. We will show that it does so with constant $n = 3$. Consider z in L_2 with $d(z) > 3(e(z) + 1)$. Then z has at least four distinguished but nonexcluded positions and we will argue in two cases.

Case 1 : z is such that a deletion of any single symbol of z results a string in L_2. In this case put v = the leftmost distinguished but nonexcluded position in z, $x = y = \epsilon$ and define u and w accordingly. Obviously, $d(uvxy) \leqslant e(u) + 1 < 3(e(uvxy) + 1)$; $d(vx) = 1$ and $e(vx) = 0$ and uv^iwx^iy is in L_2 for every $i \geqslant 0$ (the case $i = 0$ follows from our case assumption).

Case 2 : z has a symbol which when deleted will result a string not in L_2. Without losing generality suppose that this symbol is a. Then z must be of the form $z = (ab)^q a (ab)^r$ where $q,r \geqslant 0$ and $q + r$ is not a prime. If $r = 0$ then z has a distinguished but nonexcluded position other than the last a. So we put v = the leftmost distinguished but nonexcluded position in z, $x = y = \epsilon$ and u and w are defined accordingly. We then have $d(vx) = 1$ and $e(vx) = 0$, $d(uvxy) \leqslant e(u) + 1 < 3(e(uvxy) + 1)$ and pumping will produce words ending with a and hence in L_2. Suppose now that $r > 0$. Then z has substring aa and at least one distinguished but nonexcluded position outside the two adjacent a's. Let v be the leftmost distinguished but nonexcluded position outside the substring aa; let $x = y = \epsilon$ and define u and w accordingly. It is easy to check that the three conditions of GOC_l hold. Hence L_2 fits exactly in area 2.

Next consider the language $L_5 = \{ a^pb^pc^rd^r \mid 1 \leqslant p \leqslant r \} \cup \{ a^pb^qc^rd^s \mid 1 \leqslant q < p$ and $p-q \leqslant \max(r, s) \} \cup \{ a^pb^qc^rd^s \mid 1 \leqslant r < s$ and $s-r \leqslant \max(p,q) \} \cup \{ a^pb^qc^{s+1}d^s \mid p, q, s \geqslant 1 \}$. It is a standard exercise to show that L_5 is not in PL and so we will argue that L_5 satisfies GOC_l thus placing L_5 in area 5. Since the last three "parts" of L_5 are lcfl's, by lemma 2.1 it suffices to show that the first part satisfies GOC_l relative to the last three. Consider $z = a^pb^pc^rd^r$ where $1 \leqslant p \leqslant r$ with a marking that satisfies $d(z) > 4(e(z) + 1)$. Then z has at least 5 distinguished positions which are not excluded. Moreover, not all d's are in excluded positions because if they were, we would have $e(z) \geqslant r$ which implies that $4(e(z) + 1) \geqslant 4r + 4 > 2(p + r) \geqslant d(z)$. This contradicts our assumption. Similarly, not all c's are in excluded positions. Since we cannot have $p = r = 1$ there are two possibilities (i) $p = 1$ (ii) $p > 1$. In the first case, consider a word z of form $z = abc^rd^r$ where $r \geqslant 2$. Let s and t be the leftmost and the rightmost distinguished but not excluded positions of c^rd^r. If s and t are both c's, there exists some nonexcluded position in d^r, and we let $v = s$ and $x =$ the rightmost nonexcluded position in d^r. If $s = c$ and $t = d$ then let $v = s$ and $x = t$. Otherwise, when both s and t are d's, there exists some nonexcluded position in c's, and we let $v =$ the leftmost nonexcluded position in c^r and $x = t$. In all of the above cases define u, w and y accordingly. Clearly, $d(u) \leqslant e(u) + 2$, $d(y) \leqslant e(y)$, $d(vx) \geqslant 1$ and $e(vx) = 0$. Thus, $d(uvxy) \leqslant e(uvxy) + 4 \leqslant 4(e(uvxy) + 1)$ and pumping will produce words of the same form and thus in L_5. Now assume that $p > 1$. Let s and t again be the leftmost and the rightmost distinguished but not excluded positions of z. There are three subcases to be considered. The first is when $s = a$ or b. In this subcase let $v = s$ and $x =$ the rightmost nonexcluded position of d^r and define u, w and y accordingly. Pumping a and d will produce words in the second or the fourth part of L_5 whereas pumping b and d will produce words in the second or the third part of L_5. The second subcase is when $s = c$. If $p < r$ then let $v = s$ and $x =$ the rightmost nonexcluded position of d^r. Define u, w and y accordingly and pumping will produce words of the same form. Otherwise, $z = a^p b^p c^p d^p$ where $p \geqslant 2$. Then by our assumption on z, there exists some nonexcluded position in a^p. Let $v =$ the leftmost nonexcluded position of a^r, $x = t$, u, w and y are defined accordingly. If $t = c$, pumping will give words in the second or the third part of L_5; otherwise, when $t = d$ we will pump a and d and obtain words in the second or the fourth part of L_5. The third subcase is when $s = d$. If $p < r$ then let $v =$ the leftmost nonexcluded position of c^r, $x = t$ and define u, w and y accordingly. Otherwise, we argue as in the second subcase when $p = r$. Again pumping will produce words in L_5. Clearly, in every case $d(u) \leqslant e(u)$, $d(y) \leqslant e(y)$, $d(vx) \geqslant 1$ and $e(vx) = 0$. Thus, $d(uvxy) \leqslant e(uvxy) + 2 < 4(e(uvxy) + 1)$ and we conclude that L_5 is in GOL_l.

Now let $K_2 = L_3 = L_5^r$, $K_3 = L_4 = L_5^a$; by theorem 4.1 and 4.2, the languages fit the appropriate areas in the figures 6, 7 and 8. Next observe that the context-free language $L_{16} = \{ a^k b^{k+m} c^m \mid k, m \geqslant 1 \}$ fits area 16. Invoking again the results of previous section we immediately conclude that languages $M_2 = L_8 = L_{16}^r$ and $M_3 = L_{12} = L_{16}^s$ also fit their corresponding areas.

We have established the properness of the respective inclusions of figure 6 and figure 7 and are now in a position to apply corollary 2 to fill the remaining areas of figure 8. In the following definitions new languages are defined by union and we stipulate that the two languages on the right-hand side are over disjoint alphabets. Let $L_6 = L_5 \cup L_8$; $L_7 = L_5 \cup L_{12}$; $L_9 = L_2 \cup L_8$; $L_{10} = L_3 \cup L_8$; $L_{11} = L_4 \cup L_8$; $L_{13} = L_2 \cup L_{12}$; $L_{14} = L_3 \cup L_{12}$; $L_{15} = L_4 \cup L_{12}$; $L_{17} = L_2 \cup L_{16}$; $L_{18} = L_3 \cup L_{16}$; $L_{19} = L_4 \cup L_{16}$; $L_{20} = L_5 \cup L_{16}$. It is easy to check that these languages fit exactly in the areas indicated by their respective indices. We may summarize our results as follows.

Theorem 4.5. The diagrams of figures 6, 7 and 8 are correct.

Comparison between PC's and Sokolowski's Conditions (SC's)

In [49], Sokolowski formulated the condition SC which he proved to be satisfied by all context-free languages ; he gave examples of languages that do not satisfy SC but do satisfy the classical pumping condition PC of [9] thus illustrating its potential utility. In [38], Nijholt observed that SC does not provide a sufficient condition for context-freedom and Grant [24] formulated ESC, a considerable strengthening of SC.

In this section we begin the comparison between ESC and SC on one hand and GOC, OC and PC on the other. We first discuss results which are directly concerned with the conditions themselves. Then we complete the comparison, as shown in figure 9, by applying the technical results obtained previously (see also [16]).

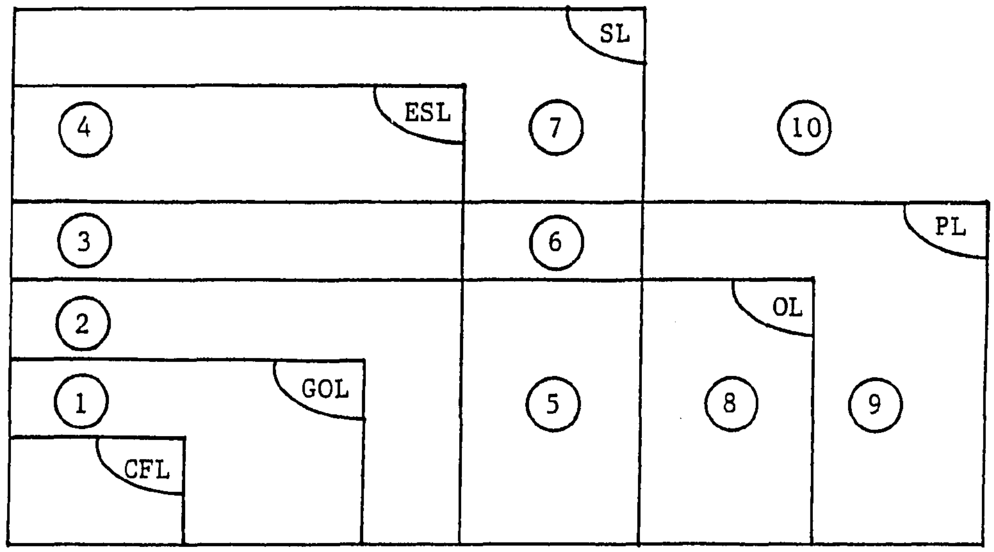

Figure 9 : Comparison between context-free pumping and Sokolowski-type conditions

Our first result is that the generalized Ogden's condition is stronger than the extended Sokolowski's condition.

Theorem 4.6. GOL $\subseteq$ ESL.

Proof. Let $L \subseteq \Sigma^*$ satisfy GOC with constant n. Let R be any unbounded relation on Σ^* and let u_1, u_2, u_3 be in Σ^* such that $\{ u_1xu_2yu_3 \mid R(x,y) \} \subseteq L$. Put $t = |u_1| + |u_2| + |u_3|$, $m = n^{t+1}$ and consider $x_1, x_2 \in \Sigma^*$ that satisfy $|x_1|, |x_2| > m$ and $R(x_1, x_2)$. Then $z = u_1x_1u_2x_2u_3$ is in L and we define a marking on z in which the positions in the u_i's are all excluded whereas the positions in x_1 and x_2 are all distinguished. We have $d(z) = |x_1| + |x_2| > 2m > n^{t+1} = n^{e(z)+1}$ and hence by GOC we may write $z = uvwxy$ so that the three conditions (of GOC) hold. There are three cases to be considered.

Case 1 : v and x both fall within x_1. Then by pumping v and x down we obtain $\hat{x}_1$ such that $\hat{x}_1 < x_1$ and $u_1\hat{x}_1u_2x_2u_3$ in L.

Case 2 : v and x both fall within x_2. By pumping v and x down we obtain $\hat{x}_2$ such that $\hat{x}_2 < x_2$ and $u_1x_1u_2\hat{x}_2u_3$ in L.

Case 3 : v falls in x_1 and x falls in x_2. By condition (2) of GOC, $d(vwx) \leqslant n^{e(vwx)+1} = n^{e(u_2)+1} \leqslant n^{t+1} = m$ and so by pumping v and x down we obtain $\hat{x}_1$ and $\hat{x}_2$ such that $(\hat{x}_1, \hat{x}_2) <_m (x_1,x_2)$ and $u_1\hat{x}_1u_2\hat{x}_2u_3$ in L.

In all cases L is shown to satisfy ESC and hence the theorem is proved. □

In [24], Grant did not consider the question of whether ESC is sufficient to guarantee context-freedom. Theorem 4.6 shows that ESC is not sufficient. We now give three examples of languages which satisfy ESC but not PC (and hence are not context-free). Our first example is $L = \{ a^r \mid r \text{ composite} \}$ and our proof will use some (possibly interesting) number-theoretic ideas. Sufficient background in number theory can be found in [5, 48].

To show that L satisfies ESC is quite easy. For any unbounded relation R and any u_1, u_2, u_3 in Σ^* such that $\{ u_1xu_2yu_3 \mid R(x,y) \} \subseteq L$ let $m = 2$. If $|x|, |y| > 2$ and $R(x,y)$ then $u_1xu_2yu_3 = a^r$ for some composite number r larger than 5. If r-1 is composite then let $\hat{x} = x$ and $\hat{y}$ is obtained from y by deleting one a. If r-1 is prime then r-2 is composite and in this case put $\hat{x} = x$ and $\hat{y}$ is obtained from y by deleting two a's. In both cases $(\hat{x},\hat{y}) <_2 (x,y)$ and $u_1\hat{x}u_2\hat{y}u_3$ is in L showing that L satisfies ESC.

To show that L does not satisfy PC let, by way of contradiction, m be the PC constant for L. We will show that there exists a composite number $n \geqslant m$ such that for every $k = 1,2,\ldots,m$ there exists an l such that $(n-k) + k \cdot l$ is a prime; this clearly contradicts our assumption. Let $p_1,p_2,\ldots,p_t$ be all the primes $\leqslant m$ and define $P = \prod_{i=1}^{t} p_i$. Consider the system of congruences

$$x \equiv 1 \pmod{p_i} \qquad (*)$$

$i = 1,2,\ldots,t$. By the Chinese remainder theorem there exists a natural number n_0 which satisfies (*) and hence so does $n_0 + q \cdot P$ for all integers q. Since $\{ n_0 + q \cdot P \mid q \geqslant 0 \}$ is an arithmetic progression there exists a q_0 such that $n = n_0 + q_0 \cdot P$ is a composite number which satisfies (*) and is $> m$. Note that $\gcd(n-k,k) = 1$ for every $k = 1,2,\ldots,m$ because if p is a prime which divides both k and n-k then $p = p_i$ for some $1 \leqslant i \leqslant t$ and this contradicts $n \equiv 1 \pmod{p_i}$. By the theorem of Dirichlet on arithmetic progressions [5, chapter 7] each of the m sequences $\{ k \cdot l + (n-k) \mid l \geqslant 1 \}$, $k = 1,2,\ldots,m$, contains infinitely many primes. This means that for every $1 \leqslant k \leqslant m$ there is an l such that $a^{n-k}(a^k)^l$ is not in L, showing that L does not satisfy PC.

Let us now consider the complement $L^c = \{ a^p \mid p \text{ prime} \}$. It is a standard exercise (e.g., [29, exercise 6.1(c)]) to show that L^c is not in PL (and hence is not context-free). To see that L^c is in ESL let R be any unbounded relation on $\{a\}^*$ and u_1,u_2,u_3 be any words in $\{a\}^*$ such that $\{ u_1xu_2yu_3 \mid R(x,y) \} \subseteq L^c$.

Recall that for any real number z, $\pi(z)$ is the number of primes $\leqslant z$. By Finsler's inequality (see for example [48, p.403]) $\pi(2n) - \pi(n) > (n/ 3 \log 2n)$

and hence $\lim_{n\to\infty} [\ \pi(2n) - \pi(n)\ ] = \infty$. It follows that

for every k there exists m_k such that $\pi(2n) - \pi(n) > k$ for all $n \geqslant m_k$ (**)

Coming back to our problem, let $k = \sum_{i=1}^{3} |u_i|$ and let m_k ($\geqslant 5$) be the corresponding constant. Define $m = m_k - 1$ and let x, y be words for which $|x|$, $|y| > m$ and $R(x,y)$. Without loss of generality suppose $|x| \leqslant |y|$. Then $w = u_1xu_2yu_3$ is in L^c, $|w| = k + |x| + |y|$ and since $k + |x| \geqslant m_k$ there are, by (**), more than k primes between $k + |x|$ and $2k + 2|x|$. We need to show that there is at least one prime between $k + |x|$ and $k + |x| + |y|$. If $2k + 2|x| > k + |x| + |y|$ then $(2k + 2|x|) - (k + |x| + |y|) = k + |x| - |y| \leqslant k$, and since there are more than k primes between $k + |x|$ and $2k + 2|x|$, and since primes cannot be consecutive there must be at least one prime between $k + |x|$ and $k + |x| + |y|$. On the other hand if $2k + 2|x| \leqslant k + |x| + |y|$ the conclusion is immediate. In both cases we may let $\hat{x} = x$ and $\hat{y}$ be obtained from y by deleting as many a's as is required to make $|u_1\hat{x}u_2\hat{y}u_3| = |u_1xu_2\hat{y}u_3|$ prime and hence $u_1\hat{x}u_2\hat{y}u_3$ in L^c. The number-theoretic argument ensured that y is long enough to allow that many deletions. It follows that L^c satisfies ESC.

The third example of a language that is in ESL but not in PL is $L = \{\ a^r \mid r \neq n! \text{ for all } n \geqslant 1\ \}$. To show that L does not satisfy PC let, by way of contradiction, $m \geqslant 2$ be the PC constant for L. Then pumping $k \leqslant m$ letters in $a^{2m!}$ yields a word of length $2m! + k(m!(m-1)/k) = (m+1)!$ which is not in L. We will omit showing that L satisfies ESC since the argument is similar to the one given above to show that the language of composite numbers satisfies ESC.

We will now employ the technical results obtained in previous section to show that none of the ten areas in figure 9 is empty thus proving the correctness of that inclusion diagram. In [7] the language $L_1 = \{ z \in \{a, b\}^* \mid z = (ab)^q \Rightarrow q$ prime $\}$ was shown to be in GOL but not in CFL. Denote by L_4 the language $\{ a^r \mid r \neq n!$ for all $n \geqslant 1 \}$; we have seen that L_4 fits exactly into area 4 in figure 9. Define $L_2 = L_4^r$ and $L_3 = L_4^a$. By theorem 4.1 L_2 is in OL and L_3 is in PL; by theorem 4.2 L_2 is in ESL but not in GOL and L_3 is in ESL but not in OL. We conclude that L_2 and L_3 fit exactly into areas 2 and 3 respectively. The language $L_7 = \{ a^n b^n c^n \mid n \geqslant 0 \}$ is not in PL, easily seen to be in SL and not in ESL [24] ; thus it exactly fits area 7 in figure 9. Putting $L_5 = L_7^r$ and $L_6 = L_7^a$ and using the previous results we can show similarly that L_5 and L_6 exactly fit the respective areas 5 and 6 of figure 9. Finally let $L_{10} = \{ xx \mid x$ in $\{ a, b, c \}^* \}$. It is again an easy exercise to show that L_{10} is not in SL and not in PL and hence in area 10. Putting $L_8 = L_{10}^r$ and $L_9 = L_{10}^a$ and using similar arguments, we obtain languages that fit exactly into areas 8 and 9 of figure 9. We summarize these results officially.

Theorem 4.7. None of the ten areas in figure 9 is empty.

One drawback of SC is that any language over one-letter alphabet satisfies SC. We now show that ESC does not share this disadvantage.

Example. We will show that the language $L = \{ a^{4^n} \mid n \geqslant 0 \}$ is not in ESL. Let $u_1 = u_2 = u_3 = \epsilon$ and let $R(x,y)$ iff $\exists k : |xy| = 4^k$ and $x = y$. Obviously R is unbounded and $\{ u_1 x u_2 y u_3 \mid R(x,y) \} \subseteq L$ (since equality holds in fact). Suppose m is the ESC constant for L ($m \geqslant 3$) and let $x = y = a^{2^{2m-1}}$. Then

$xy = a^{2^{2m-1} + 2^{2m-1}} = a^{4^m}$ is in L, R(x,y) holds and for any $1 \leqslant k < 2^{2m-1}$, $|xa^k| = 2^{2m-1} + k \neq 4^i$ for every i. Also by deleting at least one but at most m symbols from x and y we will obtain a word $\hat{x}\hat{y}$ with $|\hat{x}\hat{y}| = 4^m - k$ where $2 \leqslant k \leqslant 2m$. Clearly, $|\hat{x}\hat{y}|$ is not of the form 4^i for any i. Hence, we conclude L is not in ESL.

CHAPTER 5.

THE INTERCHANGE CONDITIONS (IC'S)

In a previous chapter we have studied and compared the various pumping conditions with respect to their power. In this chapter we compare the interchange conditions to these pumping conditions : the classic pumping condition [9], Ogden's condition [39], generalized Ogden's condition [7], linear versions of the above conditions [17], and the Sokolowski-type conditions [49, 24]. In addition we formulate an interchange condition for linear context-free languages and compare it with the other conditions.

The chapter is organized as follows. First we give the main definitions and set our terminology. We formulate the interchange conditions for context-free and linear context-free languages. Two operations on languages are defined and their relevance is indicated. In a later section we prove the interchange lemma for the linear languages. We then compare the interchange conditions with various context-free pumping conditions, linear pumping conditions and Sokolowski-type conditions, respectively. An interesting, and perhaps somewhat unexpected result here is that the interchange condition for context-free languages is strictly stronger than the Sokolowski's condition [49] while being incomparable with the extended Sokolowski's condition [24].

Defining IC's and Operations on Languages

In addition to the basic definitions introduced in previous chapters, we will present some notation and some background material for the interchange conditions [42, 37].

For a set Q we will use $\|Q\|$ to denote the *cardinality* of Q. For a language L, L^n is the set of all words of length n in L. However, when Σ is an alphabet, Σ^n denotes the set of all words of length n over Σ.

Let n, i, m be integers such that $i, m \geqslant 0$, $n \geqslant 1$, and $i+m \leqslant n$. The sequence of positions $[i+1, i+2, \cdots, i+m]$ is an *(i,m)-window* (relative to n). An *m-window* is an (i,m)-window for some $i \geqslant 0$ and a *window* is some (i,m)-window. When applying this terminology to a string of length n we identify a window with the substring that "can be seen through it". Thus for the string $z = a_1a_2 \cdots a_n$, the substring $a_{i+1}a_{i+2} \cdots a_{i+m}$ is the (i,m)-window in z. For a set of strings $R \subseteq \Sigma^n$ we define an operation of "interchanging windows" by:

$$IW_{i,m}(R) = \{ xwy \mid \exists u, v, z\, [\, |x| = |u| = i \, ; \, |w| = |z| = m \, ; \, xzy, uwv \in R] \}.$$

Example. (i) Let n = 4 and R = {abac, cabb}. Then

$IW_{0,2}(R) = \{abac, cabb, caac, abbb\}$.

(ii) Note that more generally we have $IW_{i,m}(\{z\}) = \{z\}$, $R \subseteq IW_{i,m}(R)$ and $R = IW_{0,n}(R)$ for $R \subseteq \Sigma^n$.

This operation of interchanging windows is at the heart of the *interchange condition* of [42] which we formulate next.

Interchange Condition (IC): A language $L \subseteq \Sigma^*$ satisfies IC if there exists a constant c such that for every $n \geqslant m \geqslant 2$, and for every $Q_n \subseteq L^n$ there exists a subset $R \subseteq Q_n$ for which:

(1) $\|R\| \geqslant \dfrac{\|Q_n\|}{c \cdot n^2}$ and (2) $IW_{i,k}(R) \subseteq L^n$ for some $i \geqslant 0$ and k, $m \geqslant k > \dfrac{m}{2}$.

The *Linear Interchange Condition* (IC_l) is the same as IC except that the conditions (1) and (2) are replaced by the respective linear conditions:

(1_l) $\|R\| \geqslant \dfrac{\|Q_n\|}{c \cdot n}$, and (2_l) $IW_{i,m}(R) \subseteq L^n$ for some $i \geqslant 0$.

Besides the obvious difference in the lower bounds of (1) and (1_l) note that in the linear case we can choose the exact size of the window while in the general cfl-case we can only choose bounds on the size of the window. We will denote by IL (IL_l) the class of languages that satisfy IC (IC_l). Following immediately from the definitions we have :

Corollary 1. (i) $IL_l \subseteq IL$.

(ii) Every language over a one-letter alphabet is in IL_l (and hence in IL).

To facilitate our proofs we will present two operations on languages. Their relevance to our study will be in that they will provide a systematic tool in constructing appropriate counter-examples. In chapter 4 we have defined similar operations, including the s-operation, and have supplied the necessary proofs. For completeness, we state the properties related to s-operation again here. Note that L^e is similar to the a-operation defined in chapter 4.

Let Σ be an alphabet that does not include f or g and let L be a language over Σ. Then the operations are the following.

e-operation :

$$L^e = \{ zf^n g^n \mid z \text{ in } L, n \geqslant 1 \} \cup \Sigma^*$$

s-operation :

$$L^s = \{ f^n z g^n \mid z \text{ in } L, n \geqslant 1 \} \cup \Sigma^* \qquad \text{(as in chapter 4)}$$

The first theorem shows how these operations can force languages to satisfy the pumping conditions and the second theorem allows one to locate languages that satisfy Ogden conditions but not the pumping conditions. Since the proofs are similar (and not difficult) we will leave out those that relate to pumping and present only those that relate to interchange conditions.

Theorem 5.1. For any language L (i) L^e is in PL (ii) L^s is in PL_l.

Theorem 5.2. If L is not in PL (respectively PL_l) then L^e (respectively L^s) is not in OL (respectively OL_l).

Our next result relates the e- and s-operations with the interchange conditions. We have seen that any language over a one-letter alphabet satisfies IC_l (and IC). Here we strengthen this observation.

Theorem 5.3. Let $L \subseteq \{ a \}^*$. Then L^e and L^s are in IL_l (and hence in IL).

Proof. The argument for the two cases is similar and so we will only treat the e-case, showing that L^e satisfies IC_l with constant $c = 2$. Let $n \geqslant m \geqslant 2$ and Q_n a subset of $(L^e)^n$. Define the following subsets of Q_n:

$$\begin{array}{lcl} A & = & \{ z \text{ in } Q_n \mid z_m = a \} \\ B & = & \{ z \text{ in } Q_n \mid z_m = g \} \\ C_i & = & \{ z \text{ in } Q_n \mid z = a^i f^k g^k \text{ and } z_m = f \} \end{array}$$

Note that $\|C_i\| \leqslant 1$ and that there can be at most $m - 1$ such C_i's. Let R be one of those subsets above which has the largest cardinality. Clearly $R \subseteq Q_n$ and $IW_{0,m}(R) \subseteq L^e$. Furthermore $\|R\| = \max(\|A\|; \|B\|; 1)$, implying

$$(m + 1)\cdot\|R\| \geqslant \|A\| + \|B\| + \sum_{i<m} \|C_i\| = \|Q_n\|$$

and hence $\|R\| \geqslant \dfrac{\|Q_n\|}{m + 1} > \dfrac{\|Q_n\|}{2m} \geqslant \dfrac{\|Q_n\|}{2n}$. Thus L^e satisfies IC_l. □

Linear Interchange Lemma

It was shown in [42] that CFL $\subseteq$ IL. We will now show the analogous result LIN $\subseteq$ IL_l. Since the proofs are similar to those in [42] we will just outline the arguments. Let $G = (N, T, P, S)$ be a lcfg (linear cfg) generating a language L and let $r \geqslant 2$ be the length of a longest right-hand side of any production in P. Let n and m be integers with $n \geqslant m \geqslant r$.

Lemma 5.1. For each z in L^n there exists a nonterminal A in N and a derivation of the form $S \overset{*}{\underset{G}{\Rightarrow}} wAy \overset{*}{\underset{G}{\Rightarrow}} wxy = z$ with $m \geqslant |x| > m - (r - 1)$.

Proof. Starting from the root of a derivation tree for z, keep walking down along the "nonterminal spine" until you reach a node v with at most m leaves under it. Let this node be labeled A and let x be the (terminal) word derived from A. Obviously, $m \geqslant |x|$. The nonterminal which labels the parent node of v derives a word longer than m and hence $|x| + r - 1 > m$. □

As in [42], for any $Q_n \subseteq L^n$ (where $L = L(G)$ for lcfg G) define:

$$Q_n(i,A,j) = \{\, wxy \in Q_n \mid |w| = i,\ |y| = j,\ S \overset{*}{\underset{G}{\Rightarrow}} wAy \overset{*}{\underset{G}{\Rightarrow}} wxy \,\}.$$

Lemma 5.2. For any subset $Q_n \subseteq L^n$ there exist integers $i, j \geqslant 0$ such that $m \geqslant n - i - j > m - (r - 1)$ and a nonterminal A such that

$$\|Q_n(i,A,j)\| \geqslant \frac{\|Q_n\|}{\|N\|\cdot(r-1)\cdot n}.$$

Proof. By Lemma 5.1 $Q_n = \cup Q_n(i,A,j)$ where the union is over all nonterminals A in N and all integers $i,j \geqslant 0$ for which $m \geqslant n - i - j > m - (r-1)$. Hence $\|Q_n\| = \|\cup Q_n(i,A,j)\| \leqslant \Sigma \|Q_n(i,A,j)\|$. Since the number of terms in the sum is bounded by $\|N\| \cdot (r+1) \cdot n$ the lemma follows. □

Theorem 5.4. *(Interchange Lemma for* LIN*)* $\text{LIN} \subseteq \text{IL}_l$.

Proof. Let $G = (N, T, P, S)$ be a lcfg with $r = 2$ and $L = L(G)$. Let $c = \|N\|$ and apply Lemma 5.2. See [42]. □

Comparison between IC's and PC's

In this section we will compare the interchange conditions with the pumping conditions. The final results are depicted in the inclusion diagram of figure 10 (or the (I,C)-plane of figure 12). In effect we will show that none of the fourteen "areas" of that diagram is empty.

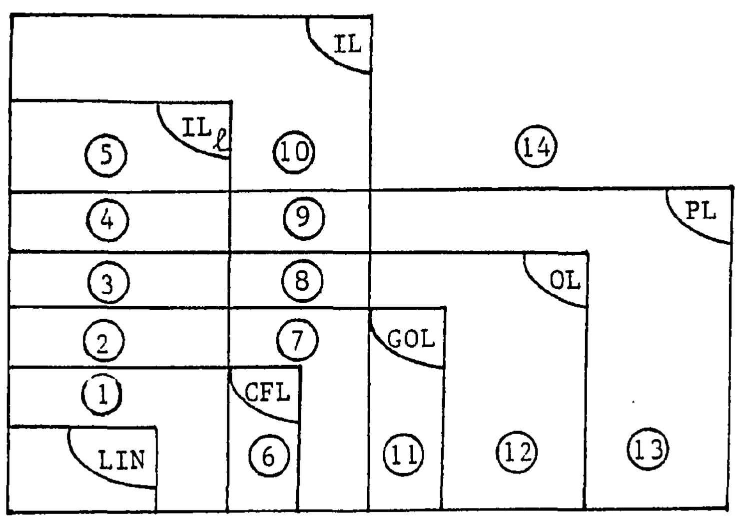

figure 10 : Relationship between Interchange and context-free pumping conditions

We now proceed to prove the correctness of the diagram in figure 10. Consider the languages: $L_1 = \{ a^m b^m c^n d^n \mid m, n \geqslant 1 \}$,
$L_2 = \{ z \text{ in } \{a,b\}^* \mid z = (ab)^q \Rightarrow q \text{ prime} \}$,
$L_3 = \{ z \text{ in } \{a,b\}^* \mid z = ab^q \Rightarrow q \text{ prime} \}$.

L_1 is in CFL but not in LIN, see [25]. L_2 was shown to be in GOL but not in CFL and L_3 was shown to be in OL but not in GOL [7]. To place L_i ($i = 1, 2, 3$) into area i of figure 10 we need to show that L_i satisfies IC_l. Since the proofs are similar we will argue only that L_2 satisfies IC_l with constant $c = 6$. Let $2 \leqslant m \leqslant n$ and $Q_n \subseteq L_2^n$. If n is odd or twice a prime, then take $R = Q_n$. Obviously $\|R\| \geqslant \frac{\|Q_n\|}{6n}$ and $IW_{0,m}(R) \subseteq L_2$. Otherwise, $n = 2t$ with t composite. Define the following subsets of Q_n:

$$
\begin{array}{lcll}
A & = & \{ z \text{ in } Q_n & \mid z \text{ ends with } a \} \\
B & = & \{ z \text{ in } Q_n & \mid z \text{ starts with } b \} \\
C & = & \{ z \text{ in } Q_n & \mid z_i = z_{i+1}, \text{ for some } i < m \} \\
D & = & \{ z \text{ in } Q_n & \mid z_i = z_{i+1}, \text{ for some } i > m \} \\
E & = & \{ z \text{ in } Q_n & \mid z_m = z_{m+1} = a \} \\
F & = & \{ z \text{ in } Q_n & \mid z_m = z_{m+1} = b \}
\end{array}
$$

Let R be one of the above sets with the largest cardinality. Obviously $IW_{0,m}(R) \subseteq L_2$ and $\|R\| \geqslant \frac{\|Q_n\|}{6} > \frac{\|Q_n\|}{6n}$ showing that L_2 is in IL_l.

Now, the language $L_5 = \{ a^p \mid p \text{ prime} \}$ is not in PL while by a corollary in a previous section L_5 satisfies IC_l; hence L_5 fits exactly in area 5 of figure 10. By theorem 5.1(i), $L_4 = L_5^e$ is in PL, while by theorem 5.2 L_4 is not in OL and by theorem 5.3 L_4 is in IL_l; thus L_4 falls exactly into area 4.

Next consider the language $L_6 = \{ xuu^R\#vv^Ry \mid x, u, v, y \text{ in } \Sigma^* \}$, where $\Sigma = \{a, b, c\}$ and w^R denotes the reverse of w. Obviously L_6 is context-free and hence it is in IL.

Claim 1. L_6 is not in IL_l.

Proof. Suppose L_6 satisfies IC_l with constant c. Let $n \geqslant 2$ be an integer that satisfies $9 \cdot 2^{2(n-1)} > 9c \cdot (4n+1) \cdot 2^{n-2}$ and let

$Q_{4n+1} = \{ uu^R\#vv^R \mid |u| = |v| = n; u_i \neq u_{i+1} \text{ and } v_i \neq v_{i+1} \text{ for } 1 \leqslant i < n \}$.

Then $Q_{4n+1} \subseteq L_6^{4n+1}$, $\|Q_{4n+1}\| = (3 \cdot 2^{n-1})^2$, and for each w in Q_{4n+1}, w does not contain a subword of the form $\alpha\alpha^R\#\beta\beta^R$ except itself. Let $m = 3n + 1$. By IC_l there exists $R \subseteq Q_{4n+1}$ and an $i \geqslant 0$ such that $IW_{i,m}(R) \subseteq L_6^{4n+1}$ and $\|R\| \geqslant \frac{\|Q_{4n+1}\|}{c \cdot (4n+1)}$. By our assumption on n, $\|R\| > 9 \cdot 2^{n-2}$. On the other hand let $r = u_1u_2u\#vv_1v_2$ and $s = x_1x_2x\#yy_1y_2$ be two strings in R such that $|u_1| = |x_1| = i$ and $|u_2u\#vv_1| = |x_2x\#yy_1| = m$; i.e., $u_2u\#vv_1$ and $x_2x\#yy_1$ can be interchanged. This means that $r' = u_1x_2x\#yy_1v_2$ and $s' = x_1u_2u\#vv_1y_2$ are also in L_6 and hence by definition of Q_{4n+1} we must have $x = (x_1x_2)^R = x_2^Rx_1^R$ and also $x = (u_1x_2)^R = x_2^Ru_1^R$ implying that $x_1 = u_1$. Similarly we must have $y_2 = v_2$. It follows that all the strings in R have an identical prefix of length i and an identical suffix of length $n - i$. This implies that for words in R we "are free" to choose at most the u_2- and v_1- parts, i.e., a total of n positions and this means that $\|R\| \leqslant 3^2 \cdot 2^{n-2}$ contradicting an earlier conclusion. This contradiction proves the claim and places L_6 in area 6 of figure 10. □

In [42] the language of repetitive strings $L_{11} = \{ uxxy \mid x \neq \epsilon$ and x, y, u in $\Sigma^* \}$, $\Sigma = \{a, b, c\}$, was shown not to be in IL. We will show that L_{11} is in GOL. Let w be in Σ^+. A position i, $1 \leqslant i \leqslant |w|$, is said to be *deletable* if the word resulting from w by deleting w_i is in L_{11}. $\delta(w)$ denotes the number of deletable positions in w.

Claim 2. For every $\beta \in \Sigma^+$, $|\beta| \geqslant 5$, $\delta(\beta) \geqslant \frac{1}{5}|\beta|$.

Proof. Follows immediately from the fact that each block of five letters has at least one deletable position while each block of six has at least two deletable positions. □

Claim 3. L_{11} satisfies GOC with constant n = 5.

Proof. Let $z = \alpha\beta\beta\gamma$ be in L_{11} with a marking that satisfies $d(z) > 5^{e(z)+1}$. Thus z has at least 6 $d\bar{e}p$'s. If there is such a position within α, let v = leftmost such $d\bar{e}p$, $x = w = \epsilon$ and define u and y accordingly. Similarly, if there is a $d\bar{e}p$ within γ let x = rightmost such $d\bar{e}p$, $w = v = \epsilon$ and define u and y accordingly. In both cases,

$$\left.\begin{array}{rcl} d(vx) & = & 1 \text{ and } e(vx) = 0 \\ d(vwx) & = & 1 < 5^{e(vwx)+1} \\ \text{and } uv^iwx^iy & \text{is in} & L_{11} \text{ for all } i \geqslant 0. \end{array}\right\} \qquad (*)$$

Now suppose that neither α nor γ has any $d\bar{e}p$'s. Then $d(\alpha\gamma) \leqslant e(\alpha\gamma)$ and $\beta\beta$ must have at least 6 $d\bar{e}p$'s. If one of these is also deletable then we let it be v, $x = w = \epsilon$ and u, y are defined accordingly. It is easy to see that the three conditions (*) hold. Finally assume that none of the $d\bar{e}p$'s is deletable. We first show that $\beta\beta$ has at least two deletable positions which are $\bar{e}p$'s (i.e., nonexcluded). Note that $|\beta\beta| \geqslant 6$. If $\beta\beta$ had at most one deletable position which is $\bar{e}p$ then $\delta(\beta\beta) \leqslant e(\beta\beta) + 1$ and hence using claim 2 $d(\beta\beta) \leqslant |\beta\beta| \leqslant 5\delta(\beta\beta) \leqslant 5(e(\beta\beta) + 1)$ which implies $d(z) = d(\beta\beta) + d(\alpha\gamma) \leqslant 5(e(\beta\beta) + 1) + e(\alpha\gamma) \leqslant 5^{e(z)+1}$ contradicting our assumption. We conclude that $\beta\beta$ has at least two deletable positions which are $\bar{e}p$'s, as well as at least six $d\bar{e}p$'s. Define v and x as the closest pair of letters within $\beta\beta$ such that (i) one of them is deletable and $\bar{e}p$ while the other is $d\bar{e}p$, and (ii) there are at least four positions between v and x. Define u, w and y accordingly. It is easy to see that such v and x always exist. Since every block of five letters has a deletable position (within itself) and since v and x are never in the same block of five letters it follows that deleting v and x creates a repetition, i.e., uwy is in L_{11}. Obviously uv^iwx^iy is in L_{11} for $i \geqslant 1$. Since w can have at most four $d\bar{e}p$'s, $d(w) \leqslant e(w) + 4$. Also, $d(vx) = 1$ and $e(vx) = 0$. Thus $d(vwx) = d(vx) + d(w) \leqslant e(w) + 5 = e(vwx) + 5 \leqslant 5^{e(vwx)+1}$ and the claim is proved. □

To complete the proof of correctness of the diagram of figure 10 we need the following:

Claim 4. All of the classes of languages GOL, OL, PL, GOL_l, OL_l, PL_l, IL and IL_l are closed under union.

Proof. For the pumping classes the proof is given in chapter 4. Let L_i satisfy IC with constant c_i, $i = 1, 2$, and let $L = L_1 \cup L_2$. For any $n \geqslant m \geqslant 2$ and any $Q_n \subseteq L^n$ let Q_n^i be $Q_n \cap L_i$, $i = 1, 2$. Since L_i satisfies IC there exists R_i such that

$\|R_i\| \geqslant \frac{\|Q_n^i\|}{c_i n^2}$, $i = 1, 2$ and

$$IW_{r,k}(R_1) \subseteq L_1 \text{ for some } r \geqslant 0, \frac{m}{2} < k \leqslant m$$

$$IW_{s,l}(R_2) \subseteq L_2 \text{ for some } s \geqslant 0, \frac{m}{2} < l \leqslant m.$$

Without loss of generality assume that $\|R_1\| \geqslant \|R_2\|$. Put $R = R_1$. Then $R \subseteq Q_n$

and $IW_{r,k}(R) \subseteq L_1 \subseteq L$ for some $r \geqslant 0$ and $\frac{m}{2} < k \leqslant m$. Also

$$\|R\| \geqslant ½(\|R_1\| + \|R_2\|) \geqslant ½(\frac{\|Q_n^1\|}{c_1 n^2} + \frac{\|Q_n^2\|}{c_2 n^2}) \geqslant \frac{\|Q_n\|}{2cn^2}$$

where $c = \max(c_1, c_2)$. Thus L satisfies IC with constant 2c. □

We can now use claim 4 to fill the remaining areas of our diagram. In the following definitions, each of the form $L_i = L_j \cup L_k$, we will assume that L_j and L_k are over disjoint alphabets. Let

$$L_7 = L_2 \cup L_6;\ L_8 = L_3 \cup L_6;\ L_9 = L_4 \cup L_6;\ L_{10} = L_5 \cup L_6;$$

$$L_{12} = L_3 \cup L_{11};\ L_{13} = L_4 \cup L_{11};\ L_{14} = L_5 \cup L_{11}.$$

It is easy to check that each language L_i , $7 \leqslant i \leqslant 14$ fits exactly into area i of our figure 10. This completes our analysis of the relationships between the interchange conditions and the (general) pumping conditions. We summarize in

Theorem 5.5. The diagram of figure 10 is correct, i.e., none of the areas in the diagram is empty.

Comparison between IC's and PC_l's

In this section we will study the relationship between interchange and linear pumping conditions. The result is shown in figure 11 which we present here with the various areas numbered for our convenience.

The language $K_1 = \{ a^p b^p c^r d^r \mid p, r \geqslant 1 \} \cup \{ a^p b^q c^r d^s \mid 1 \leqslant p < q ; r, s \geqslant 1 \}$

$$\cup \{ a^p b^q c^r d^s \mid p, q, r, s \geqslant 1 ; 0 < p-q \leqslant r+s \}$$

was shown in chapter 4 (see also [18]) to be in GOL_l but not in LIN (the latter part follows from a lemma of Greibach [26]).

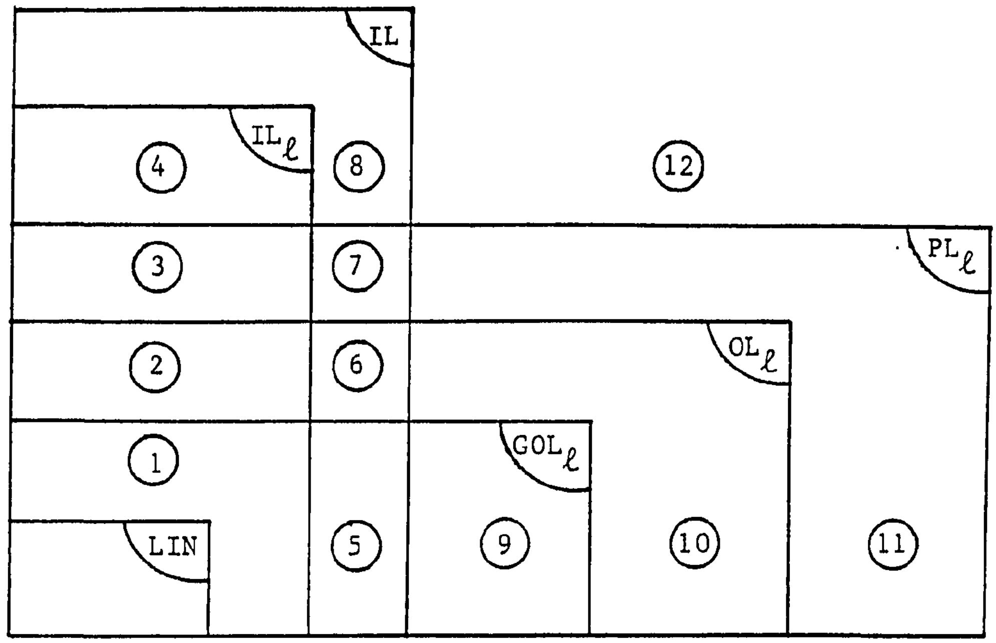

Figure 11 : Relationship between interchange and linear pumping conditions

To place K_1 in area 1 we will prove the following:

Claim 1. K_1 satisfies IC_ℓ with constant $c = 3$.

Proof. Let m, n be integers such that $n \geqslant m \geqslant 2$ and let $Q_n \subseteq K_1^n$. Define the following subsets of Q_n.

$$
\begin{array}{lcl}
A & = & \{ z \text{ in } Q_n \mid z_m = a \} \\
B & = & \{ z \text{ in } Q_n \mid z_m = d \} \\
C_i & = & \{ z \text{ in } Q_n \mid z_m = b \text{ , } \#_a(z) = i \} \\
D_j & = & \{ z \text{ in } Q_n \mid z_m = c \text{ , } \#_a(z) = j = \#_b(z) \text{ , } \#_c(z) = \#_d(z) \} \\
E & = & \{ z \text{ in } Q_n \mid z_m = c \text{ , } 1 \leqslant \#_a(z) < \#_b(z) \} \\
F_k & = & \{ z \text{ in } Q_n \mid z_m = c \text{ , } \#_a(z) + \#_b(z) = k \text{ , } 0 < \#_a(z) - \#_b(z) \leqslant \#_c(z) + \#_d(z) \}
\end{array}
$$

Note that $i \leqslant m-1$, $j \leqslant \frac{m-1}{2}$ and $k \leqslant m-1$. Let R be one of the sets above which has maximum cardinality. Then $\|R\| \cdot [3+(m-1) + \lfloor (m-1)/2 \rfloor + (m-1)] \geqslant$ $\|Q_n\|$ and so $\|R\| \geqslant \frac{\|Q_n\|}{3m} \geqslant \frac{\|Q_n\|}{3n}$. It is easy to see that $IW_{0,m}(R) \subseteq K_1$ and the claim is proved. □

Let $K_2 = \{z \text{ in } \{a,b\}^* \mid z = ab^q \Rightarrow q \text{ prime}\}$ (= L_3 of figure 10). In [18] we have shown that K_2 is in OL_l but not in GOL_l. In the previous section we have seen that K_2 is in IL_l; thus K_2 fits area 2.

Next put $K_4 = \{a^p \mid p \text{ prime}\}$ (= L_5 of figure 10). Obviously K_4 satisfies IC_l and not PC_l; hence it is in area 4. Define $K_3 = K_4^s$. By theorem 5.1 K_3 is in PL_l and by theorem 5.2 it is not in OL_l. By theorem 5.3 K_3 is in IL_l and hence it exactly fits area 3 of figure 11.

Define now $K_9 = \{uxxy \mid x \neq \epsilon \text{ and } u, x, y \text{ in } \Sigma^*\}$ (= L_{11} of figure 10); K_9 was shown not to be in IL [42, 46] and we have shown in the previous section that it is in GOL. Using similar arguments we can prove that K_9 is in GOL_l implying its placement in area 9 of figure 11. Using disjoint union and the fact that all the classes of languages discussed here are closed under union we can fill areas 10, 11 and 12 by the respective languages $K_{10} = K_2 \cup K_9$, $K_{11} = K_3 \cup K_9$ and $K_{12} = K_4 \cup K_9$.

It remains to fill the "vertical areas" 5, 6, 7 and 8. These are the areas that separate IC_l and IC and in the following discussion we will try to shed some light on the difference between those two conditions.

Let $a \geqslant 1$ and $b \geqslant 0$ be two integers and let f and g be two letters not in Σ. We define an operation on languages $L \subseteq \Sigma^*$ by

$L(a,b) = \{ f^i w g^i \mid |w| = ai + b , i \geqslant 1 , w \in L \} \cup \{ f^m x g^n \mid n \neq m , x \in \Sigma^* \}$.

The first property of this operation is that it forces languages into GOL_l.

Theorem 5.6. $L(a,b)$ is in GOL_l.

Proof. Since the second "part" of $L(a,b)$ is lcfl, we only need to consider $z = f^i w g^i$ where $w \in L$, $|w| = ai + b$ for some $i \geqslant 1$ and z has a marking such that $d(z) > k(e(z) + 1)$ where $k = \max\{ a+2, b \} + 1$. If there exist some $\bar{d}$eps among f's then let v = the leftmost $\bar{d}$ep in f^i, let $x = y = \epsilon$ and define u and w accordingly. Otherwise, if there exist some $\bar{d}$eps among g's then let x = the rightmost $\bar{d}$ep in g^i, let $u = v = \epsilon$ and define w and y accordingly. Finally, if there are no $\bar{d}$eps among f's or g's then there must exist some $\bar{e}$p among f's (otherwise we would have $e(z) \geqslant i$ which implies that $k(e(z) + 1) \geqslant k(i + 1) > (a+2)i + b = |z| \geqslant d(z)$ contradicting our assumption). Thus we let v = the leftmost $\bar{e}$p in f^i, x = the rightmost $\bar{d}$ep in w and define u, v and y accordingly. Clearly in all cases, $d(u) \leqslant e(u)$, $d(y) \leqslant e(y)$, $d(vx) = 1$, $e(vx) = 0$. Therefore, $d(uvxy) \leqslant e(uvxy) + 1 < k(e(uvxy) + 1)$ and, moreover, pumping produces words in $L(a,b)$. Thus $L(a,b)$ satisfies GOC_l. □

Next we show that IL is closed under this operation.

Theorem 5.7. If L is in IL then so is $L(a,b)$.

Proof. Let $L_2 = \{ f^m x g^n \mid n \neq m , x \in \Sigma^* \}$ and $L_1 = L(a,b) - L_2$; since L_2 is cfl it satisfies IC. Let c and c_2 be the IC-constants of L and L_2 respectively and let $n \geqslant m \geqslant 2$ be integers with $Q_n \subseteq L(a,b)^n$. Partition Q_n into two sets, $A = Q_n \cap L_1$ and $B = Q_n \cap L_2$. Since L_2 satisfies IC with constant c_2 and $B \subseteq L_2^n$, there exists a subset $R_2 \subseteq B$ for which :

(i) $\|R_2\| \geqslant \frac{\|B\|}{c_2 \cdot n^2}$ and (ii) $IW_{i,k}(R_2) \subseteq L_2^n$ for some $i \geqslant 0$ and $m \geqslant k > \frac{m}{2}$.

Note that for some n the set A is empty and for such n we let $R = R_2$. Otherwise, there are two cases to be considered.

Case 1 : $2 \leqslant m \leqslant \alpha = \frac{an+2b}{a+2}$. Put $Q_A = \{w \in L \mid f^i w g^i \in A\}$ and note that $Q_A \subseteq L^\alpha$. Since L satisfies IC with constant c there exists a subset $R_A \subseteq Q_A$ for which: (i) $\|R_A\| \geqslant \frac{\|Q_A\|}{c\alpha^2} > \frac{\|A\|}{cn^2}$, and (ii) $IW_{j,k}(R_A) \subseteq L^\alpha$ for some $j \geqslant 0$ and $m \geqslant k > \frac{m}{2}$. Define $R_1 = \{f^i w g^i \mid w \in R_A, i = \frac{n-\alpha}{2}\}$. Clearly, $R_1 \subseteq A \subseteq Q_n$ and also $IW_{l,k}(R_1) \subseteq L_1^n$ for $l = i+j$ and $m \geqslant k > \frac{m}{2}$. Furthermore $\|R_1\| = \|R_A\|$ and so $\|R_1\| = \|R_A\| > \frac{\|A\|}{cn^2}$. Now choose R to be the set with the larger cardinality among R_1 and R_2. We have $(c+c_2)\, n^2 \|R\| \geqslant cn^2\|R_1\| + c_2 n^2 \|R_2\| > \|A\| + \|B\| = \|Q_n\|$ and so the condition IC is satisfied in this case.

Case 2 : $2 \leqslant \alpha < m$. Here we can take $R_1 = A$, $l = \frac{n-\alpha}{2}$, $k = m$ and have $\|R_1\| = \|A\| > \frac{\|A\|}{cn^2}$ and $IW_{l,k}(R_1) \subseteq L_1^n$. The rest of the argument is as in case 1.

It follows that L(a,b) satisfies IC with constant $c+c_2$. □

Theorem 5.8. If L is in IL_l then so is L(a,b).

Proof. Similar to the proof of theorem 5.7. □

Recall now the language $L_6 = \{xuu^R\#vv^Ry \mid x, y, u, v \text{ in } \Sigma^*\}$ of the third section. There we have shown that L_6 is in IL but not in IL_l. Let $a > 1$ and define $K_5 = L_6(4a, 1)$. By theorem 5.6 K_5 is in GOL_l and by theorem 5.7 it is in IL. We will now show that K_5 is not in IL_l thus placing K_5 into area 5 of figure 11. Suppose K_5 satisfies IC_l with constant c. Let $n \geqslant 2$ be an integer that satisfies

$$9{\cdot}2^{2an-2} > 9c{\cdot}[(4a+2)n + 1]{\cdot}2^{(a+1)n-2} \qquad (*)$$

and denote $\sigma = (4a+2)n + 1$. Define

$Q_\sigma = \{ f^n uu^R\#vv^R g^n \mid |u| = |v| = an \,;\, u_i \neq u_{i+1} \text{ and } v_i \neq v_{i+1} \text{ for } 1 \leqslant i < an \}$.

Then $Q_\sigma \subseteq K_5^\sigma$ and $\|Q_\sigma\| = (3{\cdot}2^{an-1})^2 = 9{\cdot}2^{2an-2}$. Note that for each $f^n wg^n$ in Q_σ, w does not contain a subword of the form $xx^R\#yy^R$ except itself. Now put $m = (3a+1)n + 1$. By IC_l there exists $R \subseteq Q_\sigma$ and an $i \geqslant 0$ such that

$\|R\| \geqslant \dfrac{\|Q_\sigma\|}{c\sigma}$ and $IW_{i,m}(R) \subseteq K_5^\sigma$. By (*), $\|R\| > 9{\cdot}2^{(a+1)n-2}$. On the other hand, let $r = \alpha_1\alpha_2\#\beta_1\beta_2$ and $s = \delta_1\delta_2\#\gamma_1\gamma_2$ be two strings in R such that $|\alpha_1| = |\delta_1| = i$ and $|\alpha_2\#\beta_1| = |\delta_2\#\gamma_1| = m$; i.e., $\alpha_2\#\beta_1$ and $\delta_2\#\gamma_1$ can be interchanged without leaving K_5. This means that $r' = \alpha_1\delta_2\#\gamma_1\beta_2$ and $s' = \delta_1\alpha_2\#\beta_1\gamma_2$ are in K_5. By our choice of m, any (i,m)-window of any word in K_5^σ must contain the middle $2an + 1$ positions. Note that this implies that $0 \leqslant i \leqslant (a+1)n$. We will assume that $n < i \leqslant an$, leaving the other cases to the reader. Our assumption implies the following equalities:

$$\begin{array}{llll} \alpha_1 = f^n u_1 \,; & \alpha_2 = u_2 u \,; & \beta_1 = vv_1 \,; & \beta_2 = v_2 g^n \\ \delta_1 = f^n x_1 \,; & \delta_2 = x_2 x \,; & \gamma_1 = yy_1 \,; & \gamma_2 = y_2 g^n \end{array}$$

where $u = (u_1u_2)^R$, $v = (v_1v_2)^R$, $x = (x_1x_2)^R$ and $y = (y_1y_2)^R$. Since r' and s' are also in K_5 our choice of Q_σ implies $x = (u_1x_2)^R$ and $y = (y_1v_2)^R$ implying $u_1 = x_1$ and $v_2 = y_2$. This further implies that all strings in R have identical prefixes of length $|\alpha_1| = i > n$ and identical suffixes of length $|\beta_2| = \sigma - m - i = (a+1)n - i$. This means that for words in R we are "free to choose" at most the positions i+1 through an+n and 3an+n+2 through 3an+n+1+i, i.e., a total of at most an+n positions. Hence $\|R\| \leqslant 9 \cdot 2^{(a+1)n-2}$ contradicting an earlier conclusion. It follows that K_5 does not satisfy IC_l and hence fits area 5 of figure 11.

We can now fill the remaining areas of the diagram of figure 11 by using the disjoint union idea (see the end of previous section) and defining $K_6 = K_2 \cup K_5$, $K_7 = K_3 \cup K_5$, and $K_8 = K_4 \cup K_5$ (these will fit into areas 6, 7 and 8 respectively). We summarize the results of this section in

Theorem 5.9. The diagram of figure 11 is correct, i.e., none of the areas is empty.

At this point we can depict some of the results obtained so far in a "three-dimensional diagram" shown in figure 12. The comparison of context-free and linear pumping conditions, see chapter 4, appears here in the (C,L)-plane. The comparison between the interchange conditions and the context-free pumping conditions (linear pumping conditions) of figure 10 (figure 11) is demonstrated as the (I,C)-plane ((I,L)-plane) of figure 12.

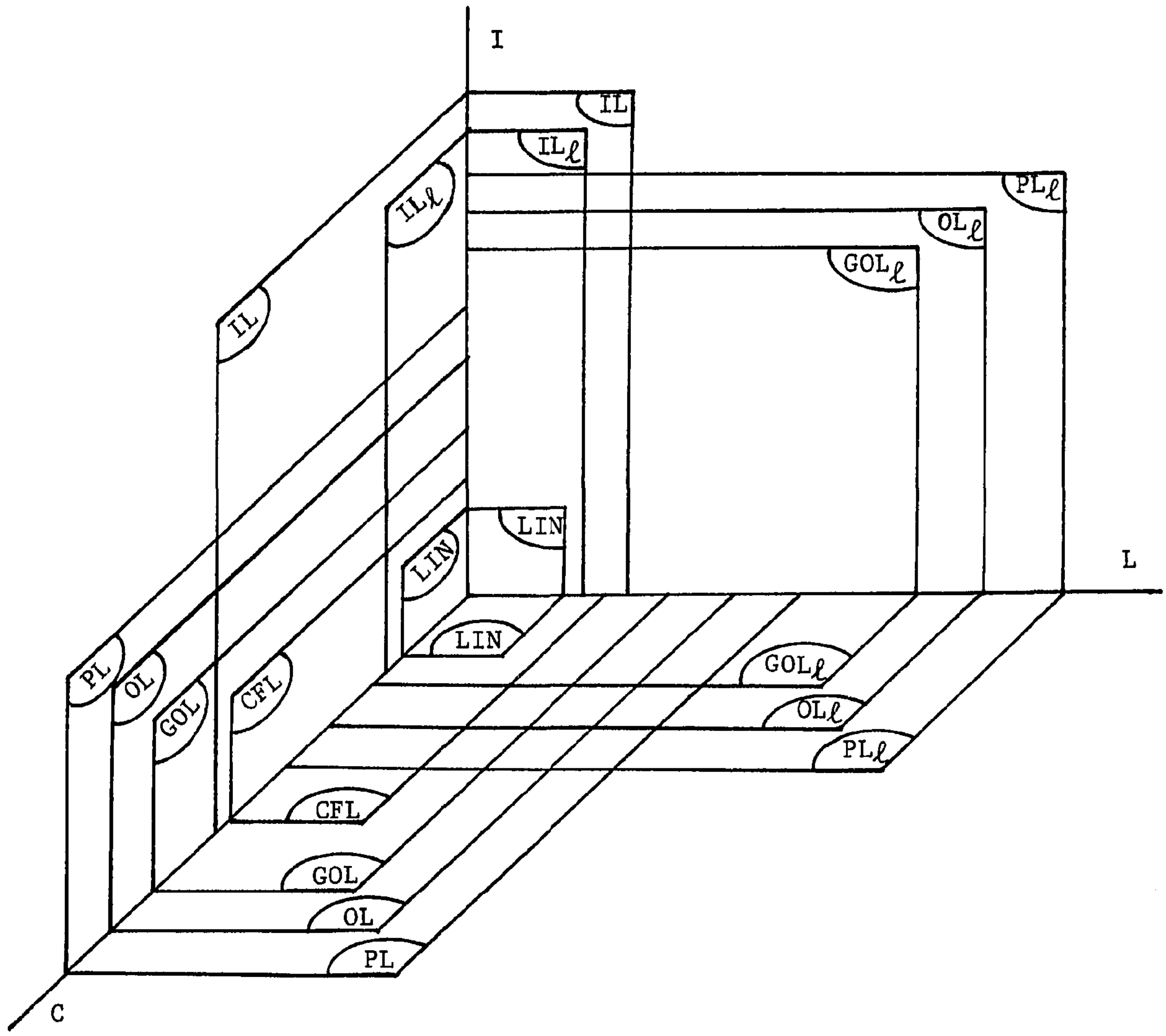

Figure 12 : Summary of the comparison among context-free pumping, linear pumping and interchange conditions

Comparison between IC's and SC's

Now we will study the relationship between IC's [46, 42] and SC's [49, 24]. The result of the comparison is shown in figure 13.

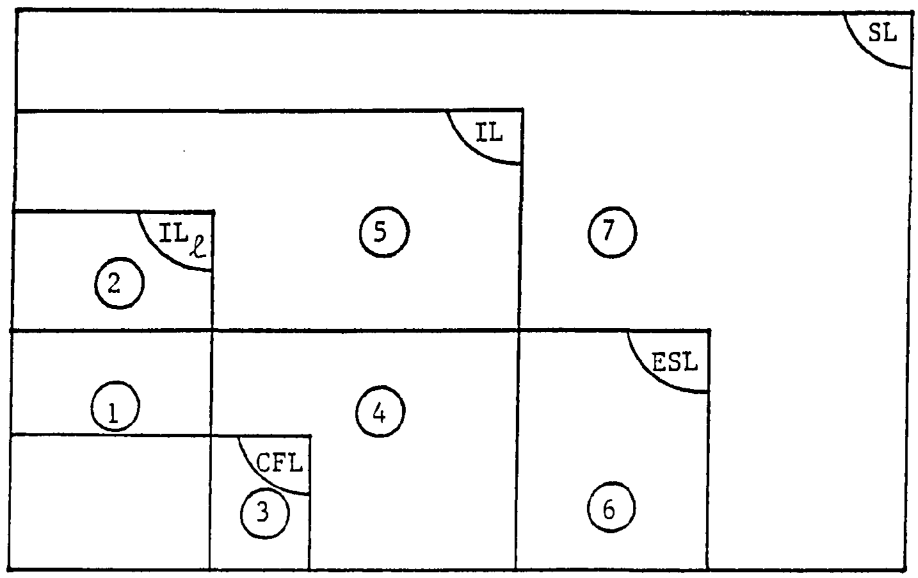

Figure 13 : Relationship between interchange and Sokolowski-type conditions

Our objective here is to show that the diagram of figure 13 is correct. We first need a combinatorial lemma.

Lemma 5.3. Let $1 \leqslant r \leqslant n$ be integers and let Δ be an alphabet with $\|\Delta\| = t$. Let the subset $Q \subseteq \Delta^n$ have the largest cardinality such that for every $0 \leqslant i \leqslant n-r$ all the (i,r)-windows in (strings of) Q are distinct. Then $\|Q\| = t^r$.

Proof. Obviously $\|Q\| \leqslant t^r$. In the other direction we can construct a set $Q \subseteq \Delta^n$ of cardinality t^r having the desired property as follows:

(a) Let $i = 0$ and $Q = \Delta^r$. Clearly $\|Q\| = t^r$ and all of the r-windows in Q are distinct. Proceed to step (b).

(b) Here all of the (j,r)-windows of Q for $0 \leqslant j \leqslant i$ are distinct. Partition Q into t^{r-1} groups, each group containing t strings with an identical $(i+1,r-1)$-window (the last $r-1$ positions). Extend the length of each string in Q by appending every string in the same group with a distinct symbol of Δ. Strings in Q are now of length $r+i+1$ and have distinct (j,r)-windows for $0 \leqslant j \leqslant i+1$. Proceed to step (c).

(c) Put $i = i+1$. If $i < n-r$ then proceed to step (b); otherwise the construction has been completed.

Q now contains only strings of length $r + (n-r-1) + 1 = n$ with all the (j,r)-windows distinct for every $0 \leqslant j \leqslant n-r$. Also, the cardinality of Q is maintained by step (b) and so $\|Q\| = t^r$. □

We now prove that the interchange condition is stronger than the Sokolowski's condition.

Theorem 5.10. $IL \subseteq SL$.

Proof. Let $L \subseteq \Sigma^*$ satisfy IC with constant c. Since, as Horvath [31] observed, every language over a two-letter alphabet satisfies SC, we may assume that $|\Sigma| \geqslant 3$. Let $\Delta \subseteq \Sigma$ with $\|\Delta\| \geqslant 2$ and u_1, u_2, u_3 in Σ^* such that

$$L_1 = \{ u_1xu_2xu_3 \mid x \in \Delta^+ \} \subseteq L.$$

We want to show that there exist x', x'' in Δ^+, $x' \neq x''$, such that $u_1x'u_2x''u_3$ is in L. Let $p = |u_1| + |u_2| + |u_3|$ and let n be an integer divisible by 8 and large enough to satisfy the inequalities $n > 8\cdot\max(|u_1|,|u_2|,|u_3|)$ and $\frac{2^{n/8}}{c(2n+p)^2} \geqslant 2$. By Lemma 5.3 there exists a subset $A \subseteq \Delta^n$, $\|A\| = 2^{n/8}$, such that for every $0 \leqslant i \leqslant \frac{7n}{8}$ all the $(i,\frac{n}{8})$-windows of A are distinct. Define $Q_{2n+p} = \{u_1xu_2xu_3 \mid x \text{ in } A\}$. Clearly, $Q_{2n+p} \subseteq L_1^{2n+p} \subseteq L^{2n+p}$ and $\|Q_{2n+p}\| = \|A\| = 2^{n/8}$. Since L satisfies IC we can take m=n and obtain a subset $R \subseteq Q_{2n+p}$ for which (i) $\|R\| \geqslant \frac{\|Q_{2n+p}\|}{c(2n+p)^2} \geqslant 2$, and (ii) $IW_{i,k}(R) \subseteq L^{2n+p}$ for some $i \geqslant 0$ and $n \geqslant k > \frac{n}{2}$. Note that for $w = u_1xu_2xu_3$ in R and for $n \geqslant k > \frac{n}{2}$ every k-window of w contains some $\frac{n}{8}$-window of x. There are several cases to be argued according to the position i of the k-window. We will discuss the case when the (i,k)-window includes u_2 and leave the other cases to the reader. Let $r = u_1x_1x_2u_2x_3x_4u_3$ and $s = u_1y_1y_2u_2y_3y_4u_3$ be two distinct strings in R such that $|x_1| = |y_1| = i-|u_1|$ and $|x_2u_2x_3| = |y_2u_2y_3| = k$. Since the length of all these strings is n we must have $|x_i| = |y_i|$, $i=1, 2, 3, 4$. By the above discussion and by the definition of Q_{2n+p}, either $x_2 \neq y_2$ or $x_3 \neq y_3$. Since we can interchange the two windows without leaving L we have $u_1x_1y_2u_2y_3x_4u_3$, $u_1y_1x_2u_2x_3y_4u_3$ in L^{2n+p}. By definition of Q_{2n+p} we have $x_1x_2 = x_3x_4$ and $y_1y_2 = y_3y_4$. Suppose now that we also have $x_1y_2 = y_3x_4$ and $y_1x_2 = x_3y_4$. Since $|x_1| < |x_3|$ is impossible we must have $|x_1| \geqslant |x_3|$ and so $x_1 = x_3v_1$ for some v_1. Hence

$$
\begin{aligned}
x_3x_4 = x_1x_2 = x_3v_1x_2 \;&\Rightarrow\; x_4 = v_1x_2 \\
&\Rightarrow\; x_1y_2 = y_3x_4 = y_3v_1x_2 \;\Rightarrow\; y_2 = x_2 \\
y_3x_4 = x_1y_2 = x_3v_1y_2 \;&\Rightarrow\; y_3 = x_3.
\end{aligned}
$$

This gives a contradiction which implies that either $x_1y_2 \neq y_3x_4$ or $y_1x_2 \neq x_3y_4$ thus providing us with an unequal pair of strings x' and x" for which $u_1x'u_2x''u_3$ is in L. This completes the proof of the theorem. □

We will now proceed to show that none of the seven areas of figure 13 is empty. Let

$$
\begin{array}{lcll}
M_1 & = & \{ a^p \mid p \text{ prime} \} & (= L_5 \text{ of figure } 10) \\
M_2 & = & \{ a^n b^n c^n \mid n \geqslant 0 \} & \\
M_3 & = & \{ xuu^R\#vv^Ry \mid x, y, u, v \text{ in } \Sigma^* \} & (= L_6 \text{ of figure } 10) \\
M_6 & = & \{ uxxy \mid x \neq \epsilon \,;\, x, y, u, \in \Sigma^* \} & (= L_{11} \text{ of figure } 10)
\end{array}
$$

where $\Sigma = \{a, b, c\}$. M_1 is in ESL − CFL [18] and M_2 is in SL − ESL [24]. Moreover, by corollary 1, M_1 is in IL_l. Since it is easy to show that M_2 is in IL_l we conclude that M_1 and M_2 fit areas 1 and 2 of figure 13. M_3 was shown to be in CFL − IL_l and M_6 in GOL − IL. Since GOL ⊆ ESL (chapter 4) M_3 and M_6 fit into areas 3 and 6 of figure 13. The remaining languages can be constructed by disjoint unions: $M_4 = M_1 \cup M_3$, $M_5 = M_2 \cup M_3$, $M_7 = M_2 \cup M_6$. Using the closure under union of all the classes involved (see claim 4 and also [18]) it is straightforward to show that M_4, M_5 and M_7 fit areas 4, 5 and 7 of figure 13. We summarize in a theorem.

Theorem 5.11. The diagram of figure 13 is correct, i.e., none of the seven areas in the diagram is empty.

CHAPTER 6.

PUMPING CONDITIONS FOR $DCFL_r$

Igarashi [32] proved three, rather interesting, pumping lemmas for real-time deterministic context-free languages and raised the natural question of the sufficiency of these conditions. In this chapter we will show that none of his pumping conditions is sufficient. We start by recalling the Igarashi's conditions, some relevant definitions and notation used.

Defining Igarashi's Pumping Conditions

We first introduce standard notation and terminology, see for example [27, 29, 32]. A *pushdown automaton* (a pda) is an acceptor with a one-way input tape, a pushdown tape, and a finite state control. It can be specified by a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ where Q is a finite set of *states* , Σ is the *input alphabet* , Γ is the *stack alphabet* , $F \subseteq Q$ is the set of *final states* and δ is a *transition function* from $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma$ to a finite subsets of $Q \times \Gamma^*$. A *deterministic pushdown automaton* (dpda) is a pda where the transition function δ has the following restrictions : For each q in Q and Z in Γ either $\delta(q, a, Z)$ contains exactly one element for all a in Σ and $\delta(q, \epsilon, Z) = \emptyset$, or $\delta(q, \epsilon, Z)$ contains exactly one element and $\delta(q, a, Z) = \emptyset$ for each a in Σ. A dpda is *real-time* iff $\delta(q, \epsilon, Z) = \emptyset$ for all q in Q and Z in Γ. It is well known that a language is context-free iff it is accepted by some pda [27, 29, 32]. A language L is *deterministic (real-time deterministic)* iff it can be accepted by some dpda (real-time dpda). For a language L over Σ, $\equiv_L$ is an equivalence relation defined on Σ^* : $x \equiv_L y$ iff for every $w \in \Sigma^*$, $xw \in L$ iff

$yw \in L$. Let $x, y \in \Sigma^*$ such that $xy \in L$. We say that " *x can be pumped in xy* " if x can be factorized $x = x_1x_2x_3$ such that $|x_2| \geqslant 1$ and for every $t \geqslant 0$, $x_1x_2^tx_3y \in L$. DCFL and $DCFL_r$ denote the classes of deterministic context-free and real-time deterministic context-free languages, respectively.

We now recall the pumping conditions of [32].

Igarashi's Condition 1 (IGC_1) : language $L \subseteq \Sigma^*$ satisfies IGC_1 if there exist constants $k_1 > 0$ and k_2, depending only on L, such that for every $n > 0$ and all x_i, $y_i \in \Sigma^*$ $(1 \leqslant i \leqslant n)$ if

(1) $i \neq j$ implies $x_i \not\equiv_L x_j$ $(1 \leqslant i, j \leqslant n)$, and

(2) for each i, $x_iy_i \in L$ and $|y_i| \leqslant (\log_2 n)/k_1 + k_2$

then there exists an r $(1 \leqslant r \leqslant n)$ such that x_r can be pumped in x_ry_r. □

Igarashi's Condition 2 (IGC_2) : language $L \subseteq \Sigma^*$ satisfies IGC_2 if there exist constants $k_1, k_2 > 0$ and k_3, depending only on L, such that for every $n > k_1$ and $m > 0$ and for all x_i, y_{ij}, w_{ij} $(1 \leqslant i \leqslant n ; 1 \leqslant j \leqslant m)$ if

(1) $r \neq s$ implies $x_iy_{ir} \not\equiv_L x_iy_{is}$ $(1 \leqslant i \leqslant n ; 1 \leqslant r, s \leqslant m)$, and

(2) $i \neq j$ implies $x_i\bar{y}_{ir} \not\equiv_L x_j\bar{y}_{js}$ for every prefix $\bar{y}_{ir}$ $(\bar{y}_{js})$ of y_{ir} (y_{js}), $(1 \leqslant i, j \leqslant n ; 1 \leqslant r, s \leqslant m)$, and

(3) $x_iy_{ij}w_{ij} \in L$ and $|w_{ij}| \leqslant (\log_2 m)/k_2 + k_3$, $(1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m)$

then there exist p, q $(1 \leqslant p \leqslant n, 1 \leqslant q \leqslant m)$ such that x_p can be pumped in $x_py_{pq}w_{pq}$. □

Let f be a function from nonnegative integers to nonnegative integers.

Non f-characteristic Condition (NC_f) : language $L \subseteq \Sigma^*$ satisfies NC_f if there exist integers n, m > 0 such that for all $x_i, y_{ij}, w_{ij} \in \Sigma^*$ ($1 \leqslant i \leqslant n$; $1 \leqslant j \leqslant m$) if conditions (1) and (2) of IGC_2 hold then there exist p ($1 \leqslant p \leqslant n$) and q ($1 \leqslant q \leqslant m$) such that [$x_p y_{pq} w_{pq} \in L$ and $|w_{pq}| \leqslant f(n) \Rightarrow x_p$ can be pumped in $x_p y_{pq} w_{pq}$].

□

Language L satisfies the *Non-characteristic Condition* (NC) if L satisfies NC_f for every function f ; note that the n and m of NC_f may depend on f. We will denote the class of languages that satisfy a certain condition by changing the letter C to letter L ; for example, IGL_1 is the class of languages that satisfy IGC_1, etc. Igarashi [32] proved the following.

Lemma 6.1. (Pumping lemma for $DCFL_r$) $DCFL_r \subseteq IGL_1$ and $DCFL_r \subseteq IGL_2 \subseteq NL$ □

Examples of languages in DCFL-NL and IGL_1-IGL_2 were also given in [32].

Languages Satisfying Igarashi's Conditions

To prove insufficiency of the Igarashi's pumping conditions we define a language-theoretic operation which forces languages to satisfy all those conditions. This operation, which we shall denote by π, was introduced in [21]. Let $\Delta = \{ 1, 2 \}$ and $\Sigma = \{ a_{ij} \mid 0 \leqslant i, j \leqslant 3 \}$. Define two functions $f_i : \Sigma \to \Sigma$ ($i = 1, 2$) by : $f_1(a_{ij}) = a_{mj}$, $f_2(a_{ji}) = a_{jm}$ where $m = i+1 \pmod 4$. A *legal string* over Σ is any string $x = \sigma_1^{n_1} \cdots \sigma_m^{n_m}$ ($m \geqslant 1$) with $\sigma_1 = a_{00}$, $n_i \geqslant 1$ and σ_{i+1} being either $f_1(\sigma_i)$

or $f_2(\sigma_i)$ ($1 \leqslant i < m$). Each legal string x defines a corresponding sequence of f_i's and the resulting string of subscripts (of the f_i's) forms the *coded string* y associated with x. For example, $x = a_{00}a_{10}^3a_{11}a_{12}^2a_{13}a_{10}^5$ is a legal string with associated coded string y = 12222. The *parity* of a string (over Σ) is the sum, modulo 2, of all the subscripts i, j ; thus the parity of the string x above is 0+3+2+6+4+5 (mod 2) = 0. Let $L \subseteq \Delta^*$. Define $\pi(L) = \{ x \in \Sigma^+ \mid x$ is legal and codes some $y \in L \} \cup \{ x \in \Sigma^+ \mid x$ is illegal and has parity $0 \}$.

The following theorem provides the main tool.

Theorem 1. For any language $L \subseteq \Delta^*$, $\pi(L)$ satisfies IGC_1, IGC_2 and NC.

Proof. We will show that $\pi(L)$ satisfies IGC_2. Let $k_1 = \sum_{i=0}^{4} 16^i$, $k_2 = 1$ and $k_3 = 0$. Let $n > k_1$, m be integers and x_i, y_{ij}, w_{ij} ($1 \leqslant i \leqslant n$; $1 \leqslant j \leqslant m$) be strings satisfying conditions (1), (2) and (3) of IGC_2. Then there exists x_p such that $|x_p| \geqslant 5$ (otherwise, there are at most k_1 distinct x_i's; since $n > k_1$, there exist $x_i = x_j$ for some $i \neq j$ which contradicts condition (2) of IGC_2). Consider this x_p in two cases.

Case 1 : There exists q ($1 \leqslant q \leqslant m$) such that $x_py_{pq}w_{pq}$ is legal. If x_p has a doublet $\sigma\sigma$, we have $x_p = x_1x_2x_3$ where $x_2 = \sigma$ (any of $\sigma\sigma$). Then for all $i \geqslant 0$, $x_1x_2^ix_3y_{pq}w_{pq}$ is legal and codes the same string that $x_py_{pq}w_{pq}$ does. Otherwise, we have $x_p = x_1x_2x_3$ where $x_2 = a_{00}$ (the second symbol of x_p) if $x_py_{pq}w_{pq}$ has parity 0 (1). Then for all $i > 0$, $x_1x_2^ix_3y_{pq}w_{pq}$ is legal and codes the same string that $x_py_{pq}w_{pq}$ does. For $i = 0$, $x_1x_3y_{pq}w_{pq}$ has parity 0 and is illegal.

Case 2 : $x_py_{pq}w_{pq}$ is illegal and has parity 0. This must be because either the initial symbol is not a_{00} or the word has bad transition. In any case $x_p y_{pq} w_{pq}$ contains a substring u of length $\leqslant 2$ such that preserving u implies preserving illegality. Since $|x_p| \geqslant 5$, there exists a substring v of x_p of length 2 which is disjoint from u. Let $x_p = x_1 x_2 x_3$ where x_2 is a nonempty substring of v of parity 0. There must be such x_2 consisting of one or two symbols. For if v has one symbol of parity 0, then x_2 can be that symbol. Otherwise v of length 2 will have parity 0 and $x_2 = v$. x_1 and x_3 are defined accordingly. Then for all $i \geqslant 0$, $x_1 x_2^i x_3 y_{pq} w_{pq}$ is illegal and has the same parity as $x_p y_{pq} w_{pq}$ which is 0.

In both cases, there exist p ($1 \leqslant p \leqslant n$) and q ($1 \leqslant q \leqslant m$) such that x_p can be pumped in $x_p y_{pq} w_{pq}$. Hence $\pi(L)$ is in IGL_2. Since $IGL_2 \subseteq NL$, L satisfies NC. Similarly we can show that $\pi(L)$ satisfies IGC_1 with constants $k_1 = 1$ and $k_2 = -\log_2(\sum_{i=0}^{4} 16^i) - 1$. □

To construct the appropriate counterexamples we need the following theorem whose proof can be obtained by simple constructions.

Theorem 2. Let $L \subseteq \Delta^*$ and let Y be one of the classes of languages CFL, DCFL and $DCFL_r$. Then $L \in Y$ iff $\pi(L) \in Y$. □

Let $L_1 = \{ 1^i 2^j 1 2^k 1^i \mid i \geqslant 1, j \geqslant k \geqslant 1 \}$. L_1 is in DCFL but not in NL [32] (and so not in $DCFL_r$). By theorems 1 and 2, $L_4 = \pi(L_1)$ is in IGL_2 (as well as IGL_1) and in DCFL but not in $DCFL_r$. Thus the answer to Igarashi's question regarding the sufficiency of IGC_2 is in the negative. Let $L_2 = \{ 1^m 2 1^n 2 1^k \mid k = m \text{ or } k = n \}$ and $L_3 = \{ 1^p \mid p \text{ prime} \}$. It is easy to see that neither L_2 nor L_3 satisfies IGC_2 and that L_2 belongs to CFL - DCFL while L_3 is not in CFL. Putting $L_5 = \pi(L_2)$ and $L_6 = \pi(L_3)$ we obtain an inclusion diagram of figure 14 where the circled number i indicates that the language L_i fits exactly that area of the diagram ($1 \leqslant i \leqslant 6$).

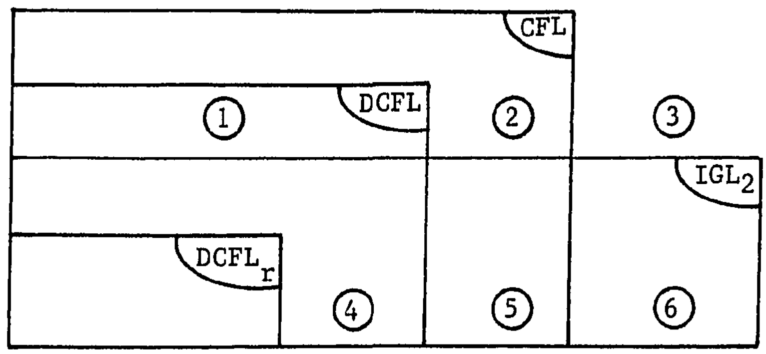

Figure 14 : An inclusion diagram for IGL_2

Now let $K_2 = \{ 1^i2^i1^j , 1^i2^j1^j \mid i, j \geqslant 1 \}$. K_2 is in CFL but not in DCFL [27] and by taking $x_i = 1^i2^i$, $y_i = 1$ ($1 \leqslant i \leqslant n$) we can show that K_2 is not in IGL_1. By theorem 1 and 2, $K_5 = \pi(K_2)$ is in CFL and IGL_1 but not in DCFL. Let $K_3 = \{ 1^p \mid p \text{ prime} \}$ as before. K_3 is not context-free and does not satisfy IGC_1. Taking $K_6 = \pi(K_3)$, by theorem 1 and 2, K_6 is in IGL_1 and still not context-free. Let $K_4 = \{ 1^i2^j1^k2^i \mid j \geqslant k \geqslant 1, i \geqslant 1 \}$. In [32], K_4 was shown to be in IGL_1 - IGL_2 and also in DCFL - $DCFL_r$. It follows that language K_i fits area i in figure 15. In this case however we have not been able to find a language that fits area 1.

Despite the fact that IGC_2 turned out to be an insufficient condition we do not yet have a clear and exact picture of the relative strength of the three conditions IGC_1, IGC_2 and NC. We thus formulate the following questions.

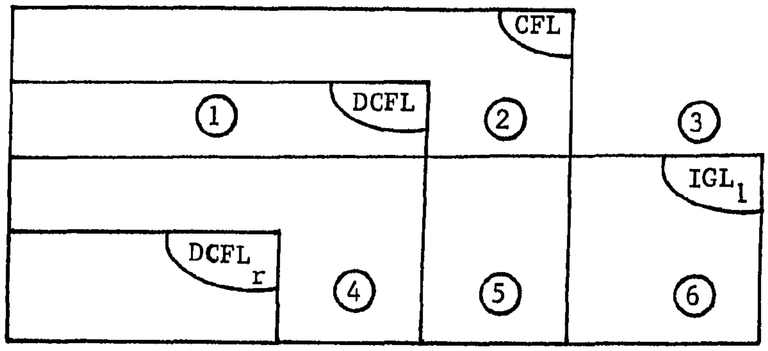

Figure 15 : An inclusion diagram for IGL_1

(1) Are IGC_2 and NC equivalent ? We know [32] that IGC_2 implies NC.

(2) Is DCFL - IGL_1 empty ?

(3) Is $IGL_2 \subseteq IGL_1$? We know that there are languages in IGL_1 - IGL_2 [32].

(4) Let $\{f_k\}$ be an increasing sequence of functions, i.e., $f_{k+1}(n) > f_k(n)$ for all n ($k \geqslant 1$). Is the corresponding hierarchy of classes of languages $\{NL_{f_k}\}$ a proper hierarchy ? It can be readily seen that, $f < g$ implies $NL_g \subseteq NL_f$.

CHAPTER 7.

CONCLUSION

This research was partly motivated by the problem of finding an "easily applicable characterization" for the class of context-free languages. Since an intercalation property gives a necessary condition for a language that belongs to a certain class, it is natural to think of an intercalation property as a bridge that might lead us to solve the problem. The main contribution of this research is the study on the relationships among various intercalation properties of context-free languages. Our results are not only the comparisons among existing intercalation properties but also include the formulation of the iteration properties of some subclasses of context-free languages. Perhaps one of the nicer features of our results is in the proof technique that used language operations giving a quick and systematic way to obtain the relationship among these various intercalation properties.

In chapter 3, we proved an Ogden-type lemma for nonterminal bounded languages. The lemma is a generalization of an Ogden's lemma for linear languages. As a consequence we also obtained a classical-type of pumping lemma for nonterminal bounded languages. It was shown that both Ogden-type conditions and the classical-type pumping conditions are not sufficient. We constructed counterexamples at various levels of the Chomsky hierarchy, each of which satisfies the conditions of our pumping lemma (Ogden-type lemma, respectively).

In chapters 4 through 6, we studied various intercalation properties of context-free languages. Among the intercalation properties that context-free languages have are the general pumping conditions, Sokolowski-type conditions and an interchange condition. The general pumping conditions consist of three types of conditions; a classical pumping condition, a stronger condition known as Ogden's condition and a generalized Ogden's condition which is the strongest condition among the three. On the other hand, of the Sokolowski-type conditions we have only two : Sokolowski's condition and an extended Sokolowski's condition. It is shown that Sokolowski's condition can be applied in some cases that the classical pumping lemma failed. In the general pumping conditions new strings in the language are obtained by "standard pumping" and deletion of subwords whereas in the interchange lemma new words are produced by subword-interchanging between words in the language. For linear context-free languages, there have been developed specialized pumping conditions of the classical pumping type [11, 29]. For this class, we have formulated the Ogden-type, the generalized Ogden's conditions and the interchange condition.

The question that we considered is how these various properties are related and how close they are to being sufficient for context-freedom. Are there context-free languages that do not satisfy any of these linear pumping conditions ? Does a language that satisfies a linear pumping condition necessarily satisfies the corresponding general pumping condition ?

In chapter 4, we answered these and many other questions regarding the relationship among various intercalation properties for context-free languages and the relationship between linear and nonlinear (i.e., general) pumping conditions. In chapter 5, we have seen three comparisons each of which was between the interchange conditions and the general pumping conditions, the linear pumping conditions and the Sokolowski-type conditions. In proving such relationships some operations on languages were introduced and their properties related to the class of language under consideration were investigated. This allowed a quick and systematic way to prove the correctness of the various inclusion relationships.

As a result, firstly we have shown that the linear and the general pumping conditions are independent (as shown in figure 8). We can see that a language that satisfies the (most stringent) generalized linear Ogden's condition need not be linear, even if it is context-free. Moreover, simultaneous satisfaction of both linear and general pumping conditions does not force a language to satisfy any stronger pumping condition, or be (linear) context-free.

Secondly, it was shown that the classical pumping properties and the Sokolowski-type conditions are incomparable except for the generalized Ogden's condition which is stronger than the extended Sokolowski's condition (as shown in figure 9). This implies that the extended Sokolowski's condition is not sufficient. In fact, we proved that there are languages that satisfy the extended Sokolowski's condition but not the classical pumping condition.

Finally, we explored the relationships related to the interchange conditions. The comparison between the interchange conditions and various pumping conditions for context-free was presented in figure 10 or the (I,C)-plane of figure 12. A similar comparison with the linear pumping conditions was shown in figure 11 or the (I,L)-plane of figure 12 and the comparison between interchange conditions and Sokolowski-type conditions was given in figure 13. We can see that the interchange condition for context-free languages is strictly stronger than the Sokolowski's condition while being incomparable with the extended Sokolowski's condition.

In chapter 6, we discussed Igarashi's pumping conditions for real-time deterministic context-free languages. We answered an open problem, formulated by Igarashi, regarding the sufficiency of these three conditions. Indeed we have shown that none of them is sufficient. Our discussion ended by suggesting future research related to the relative strength of the above three conditions.

In conclusion, in this thesis we have developed several intercalation conditions for some subclasses of context-free languages and also presented a rather complete comparison of these conditions, especially for the full class of context-free languages.

BIBLIOGRAPHY

1. Aho, A. V. and Ullman, J. D. (1972). *The Theory of Parsing, Translation, and Compiling.* Vol.1. Englewood Cliffs : Prentice-Hall.

2. Aho, A. V., Sethi, R. and Ullman, J. D. (1986). *Compilers : Principles, Techniques, and Tools.* Reading : Addison-Wesley.

3. Altman, E. B. (1964). "The concept of finite representability." SRC Report 56-A-64-20. Case Institute of Technology, Cleveland, Ohio.

4. Altman, E. B. and Banerji, R. B. (1965). "Some problems of finite representability." *Information and Control* 8 : 251-263.

5. Apostol, T. M. (1976). *Introduction to Analytic Number Theory.* New York : Springer-Verlag.

6. Autebert, J. M., Beauquier, J., Boasson, L. and Nivat, M. (1979). "Quelques problemes ouverts en theorie des languages algebriques." *RAIRO Informatique Theorique* 13(4) : 368-378.

7. Bader, C. and Moura, A. (1982). "A Generalization of Ogden's Lemma." *Journal of the Association for Computing Machinery* 29(2) : 404-407.

8. Banerji, R. B. (1963). "Phrase structure languages, finite machines and channel capacity." *Information and Control* 6 : 153-162.

9. Bar-Hillel, Y. Perles, M. and Shamir, E. (1961). "On formal properties of simple phrase structure grammars." *Z. Phonetik. Sprachwiss. Kommunikationsforch.* 14 : 143-172.

10. Beaty, J. C. (1980). "Two iteration theorems for the LL(k) languages." *Theoretical Computer Science* 12 : 193-228.

11. Berstel, J. (1979). *Transductions and Context-free Languages.* Stuttgart : B. G. Teubner.

12. Boasson, L. (1973). "Two iteration theorems for some families of languages." *Journal of Computer and System Sciences* 7(6) : 583-596.

13. Boasson, L. and Horvath, S. (1978). "On languages satisfying Ogden's Lemma." *RAIRO Informatique Theorique* 12(3) : 201-202.

14. Book, R. V. (editor). (1980). *Formal Language Theory, Perspectives and Open Problems.* New York : Academic Press.

15. Boonyavatana, R. and Slutzki, G. (1985). "Ogden's Lemma for Nonterminal Bounded Languages." TR 85-6. Dept. of Computer Science, Iowa State University, Ames, Iowa. To appear in *RAIRO Informatique Theorique.*

16. Boonyavatana, R. and Slutzki, G. (1985). "Pumping conditions for Context-free Languages - Comparison." TR 85-12. Dept. of Computer Science, Iowa State University, Ames, Iowa.

17. Boonyavatana, R. and Slutzki, G. (1985). "A Generalized Ogden's Lemma for Linear Context-free Languages." *EATCS Bulletin* 28 : 20-26.

18. Boonyavatana, R. and Slutzki, G. (1985). "Comparisons between some pumping condition for context-free languages." TR 85-32. Dept. of Computer Science, Iowa State University, Ames, Iowa.

19. Boonyavatana, R. and Slutzki, G. (1986). "The Interchange or Pump (Di) lemmas for context-free languages." TR 86-3. Dept. of Computer Science, Iowa State University, Ames, Iowa.

20. Ehrenfeucht, A. and Rozenberg, G. (1975). "A pumping theorem for deterministic ETOL languages." *RAIRO Informatique Theorique* 9 : 13-23.

21. Ehrenfeucht, A., Parikh, R. and Rozenberg, G. (1981). "Pumping lemmas for regular sets." *SIAM Journal on Computing* 10(39): 536-541.

22. Ginsburg, S. (1966). *The Mathematical Theory of Context-Free Languages.* New York : McGraw-Hill.

23. Ginsburg, S. and Spanier, E. H. (1966). "Finite turn pushdown automata." *SIAM Journal on Computing* 4(3) : 429-453.

24. Grant, P. W. (1982). "Extensions of Sokolowski's Theorem to Prove Languages are not Context-free or not Regular." *International Journal of Computer Mathematics* 11 : 187-196.

25. Greibach, S. A. (1966). "The unsolvability of the recognition of linear context-free languages." *Journal of the Association for Computing Machinery* 13(4) : 582-587.

26. Greibach, S. A. (1979). "Linearity is Polynomially Decidable for Realtime Pushdown Store Automata." *Information and Control* 42 : 27-37.

27. Harrison, M. A. (1978). *Introduction to Formal Language Theory.* Reading : Addison-Wesley.

28. Harrison, M. A. and Havel, I. M. (1974). "On the parsing of deterministic

languages." *Journal of the Association for Computing Machinery* 21(4) : 525-548.

29. Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to Automata Theory, Languages and Computation.* Reading : Addison-Wesley.

30. Horváth, S. (1978). "The family of languages satisfying Bar-Hillel's lemma." *RAIRO Informatique Theorique* 12(3) : 193-199.

31. Horváth, S. (1983). "A comparison of Iteration Conditions on Formal Languages." *Proc. Colloq. Algebra, Combinatorics and Logic in Computer Science. Colloquia Mathematica Societatis János Bolyai* , Vol. 42. pp.360-368. Eds. Demetrovics, J., Katona, Gy. and Salomaa, A. Budapest : J. Bolyai Math. Soc. and Amsterdam : North-Holland.

32. Igarashi, Y. (1985). "A pumping lemma for real-time deterministic context-free languages." *Theoretical Computer Science* 36 : 89-97.

33. Jaffe, J. (1978). "A necessary and sufficient pumping lemma for regular languages." *ACM SIGACT News* 10(2) : 48-49.

34. King, K. N. (1980). "Iteration Theorems for Families of Strict Deterministic Languages." *Theoretical Computer Science* 10 : 317-333.

35. Kløve, T. (1977). "Pumping Languages." *International Journal of Computer Mathematics* 6(2) : 115-125.

36. Krevner, Y. and Yehudai, A. (1982). "An Iteration theorem for simple precedence languages." In *Automata, Languages and Programming, Lecture Notes in Computer Science,* Vol. 140. pp. 360-368. Eds. Nielsen, M. and Schmidt, E.M. Berlin : Springer-Verlag.

37. Main, M. G. (1982). "Permutations are not Context-free : An application of the Interchange lemma." *Information Processing Letters* 15(2) : 68-71.

38. Nijholt, A. (1982). "A note on the Sufficiency of Sokolowski's Criterion for Context-free languages." *Information Processing Letters* 14(5) : 207.

39. Ogden, W. (1968). "A helpful result for proving inherent ambiguity." *Mathematical System Theory* 2(3) : 191-194.

40. Ogden, W. (1968). "Intercalation theorems for pushdown store and stack languages." Ph.D. dissertation, Stanford University.

41. Ogden, W. (1969). "Intercalation theorems for stack languages." *Proc. First annual ACM STOC :* 31-42.

42. Ogden, W., Ross, R. J. and Winklmann, K. (1985). "An 'Interchange Lemma' for Context-free languages." *SIAM Journal on Computing* 14(2) : 410-415.

43. Parikh, R. J. (1966). "On context-free languages." *Journal of the Association for Computing Machinery* 13 : 570-581.

44. Rabin, M. O. and Scott, D. (1959). "Finite automata and their decision problems." *IBM Journal of Research and Development* 3(2) : 114-125.

45. Rosenkrantz, D. J. and Stearns, R. E. (1970). "Properties of deterministic top-down grammars." *Information and Control* 17(3) : 226-256.

46. Ross, R. J. and Winklmann, K. (1982). "Repetitive strings are not Context-free." *RAIRO Informatique Theorique* 16(3) : 191-199.

47. Salomaa, A. (1973). *Formal Languages.* New York : Academic Press.

48. Sierpinski, W. (1959). *Number Theory, Part 2.* Vol. 38. Warsaw : PWN.

49. Sokolowski, S. (1978). "A method for proving programming languages non context-free." *Information Processing Letters* 7(3) : 151-153.

50. Wise, D. S. (1976). "A strong pumping lemma for context-free languages." *Journal of Computer and System Sciences* 3(3) : 359-370.

51. Yeh, R. T. (editor). (1976). *Applied Computation Theory : Analysis, Design, Modeling.* Englewood Cliffs : Prentice-Hall.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Giora Slutzki, for his guidance, patience, and most of all for the time he has spent on me over the past few years. I am grateful to him for showing me an example of an enthusiastic researcher, a kind advisor, and an excellent teacher. His constant support and dedication is deeply appreciated.

I would also like to express my gratitude to my parents who always encourage me. Without their longstanding support, I would never have come this far.